National Aeronautics and
Space Administration

NASA CR 80-10226
160658 JSC-13821
Revision A

Lyndon B. Johnson Space Center
Houston. Texas 77058

MAY 1 1980

EARTH OBSERVATIONS DIVISION

SPACE AND LIFE SCIENCES DIRECTORATE

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY FOR APPLICATIONS OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER GUIDE FOR THE IBM 370/148

VOLUME III – AS-BUILT DOCUMENTATION
(SECTIONS 1 THROUGH 12)

Job Order 76-662

Prepared By

Lockheed Engineering and Management Services Company, Inc.
Houston, Texas

Contract NAS 9-15800

(E80-10226) EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY FOR APPLICATIONS OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER GUIDE FOR THE IBM 370/148. VOLUME 3: AS-BUILT (Lockheed Engineering and   G3/43   N80-27775   Unclas 00226

April 1980

LEMSCO-12565
Revision A

1. Report No. JSC-13821, Revision A	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Earth Observations Division Version of the Laboratory for Applications of Remote Sensing System (EOD-LARSYS) User Guide for the IBM 370/148, Volume III – As-Built Documentation		5. Report Date April 1980
		6. Performing Organization Code SF4
7. Author(s) M. L. Burnell and P. J. Aucoin Lockheed Engineering and Management Services Company, Inc.		8. Performing Organization Report No. LEMSCO-12565, Revision A
9. Performing Organization Name and Address Lockheed Engineering and Management Services Company, Inc. 1830 NASA Road 1 Houston, Texas 77058		10. Work Unit No.
		11. Contract or Grant No. NAS 9-15800
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Lyndon B. Johnson Space Center (JSC) Houston, Texas 77058 (J. M. Sulester, Tech. Monitor)		13. Type of Report and Period Covered User Guide
		14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This document presents the subprogram-by-subprogram structure of the EOD-LARSYS as programmed on the Purdue University IBM 370/148 computer. It documents 182 processor subprograms and 60 utility subprograms in the system. EOD-LARSYS is the JSC version of an integrated batch system for analysis of multispectral scanner imagery data. This volume III is designed for use with the System Overview (volume I), the User's Reference Manual (volume II), and the Program Listings (volume IV). The system is operational from remote terminals at JSC under the Virtual Machine/Conversational Monitor System environment.

17. Key Words (Suggested by Author(s)) Classification Clustering Display Feature selection Histograms Landsat imagery Mapping Scatter plots Transformations	18. Distribution Statement

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 793	22. Price*

JSC-13821
Revision A

EARTH OBSERVATIONS DIVISION VERSION OF THE LABORATORY FOR APPLICATIONS OF REMOTE SENSING SYSTEM (EOD-LARSYS) USER GUIDE FOR THE IBM 370/148

VOLUME III — AS-BUILT DOCUMENTATION

Job Order 76-662

PREPARED BY

M. L. Burnell
Technical Publications Department

and

P. J. Aucoin
Earth Observations Data Products Department

APPROVED BY

NASA

[illegible]
J. M. Sulester, Technical Monitor, Systems and Facilities Branch

LOCKHEED

T. F. Mackin
T. F. Mackin, Supervisor Exploratory Investigations Section

Prepared By

Lockheed Engineering and Management Services Company, Inc.

For

Earth Observations Division

Space and Life Sciences Directorate

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

April 1980

LEMSCO-12565
Revision A

PREFACE

The system which is the subject of this documentation was designed originally for execution on the Univac 1108/1110 computer at the Laboratory for Applications of Remote Sensing, Purdue University. This volume III documented the conversion in 1978 of the EOD-LARSYS software for execution on the IBM 370/148. In 1979, the IBM 370/148 was replaced by the IBM 3031 computer, which is thoroughly compatible with the software as altered for execution on the IBM 370/148. Thus, no conversion of software is required for this system to be operable on the IBM 3031 computer. This revision primarily documents enhancements and additions to the system.

CONTENTS

TABLES

FIGURES

1. SCOPE

This document is one of a four-volume series entitled "Earth Observations Division Version of the Laboratory for Applications of Remote Sensing System (EOD-LARSYS) User Guide for the IBM 370/148." The four volumes are:

Volume I	— System Overview
Volume II	— User's Reference Manual
Volume III	— As-Built Documentation
Volume IV	— Program Listings

This volume III, As-Built Documentation, allows the user to view the system on a subprogram-by-subprogram basis. The capabilities and limitations of the system as a whole and the information required to execute each processor and obtain the desired output are stated in volume II.

Two hundred forty-two subprograms comprise the EOD-LARSYS. In this document, each subprogram is grouped according to the processor which utilizes it. Subprograms which are called by more than one processor are documented in a section dedicated to utility subprograms. Table 1-1 lists the EOD-LARSYS subprograms in alphabetical order, along with the processor to which each belongs and the section in which it is documented; utility subprograms are so designated.

In compiling the subprogram descriptions, several documents were used, with much of the material copied verbatim; these documents are listed in section 2. A description of the EOD-LARSYS monitor routines, MONTOR and MONPAC, is given in section 3, followed by a description of their supervisory routine, MSCAN (section 4), and a description of the common blocks and block data associated with the system (section 5). The processors, by section, are as follows.

Section	Processor
6	One-Dimensional Histogram (HIST)
7	GRAYMAP
8	Statistics (STAT)
9	Iterative Self-Organizing Clustering System (ISOCLS)
10	Feature Selection (SELECT)
11	Classification (CLASSIFY)
12	Performance Display (DISPLAY)
13	Data-Transformation (DATA-TR)
14	Statistics Transformation (TRSTAT)
15	N-Dimensional Histogram (NDHIST)
16	Scatter Plot (SCTRPL)
17	Dot Data (DOTDATA)
18	Automatic Cluster Labeling (LABEL)
20	Data Merge (DAMRG)
21	Ground Truth Dot Dump (GTDDM)
22	Ground Truth Tape Conversion (GTTCN)
23	Iterative Self-Organizing Clustering System Using Packed Pixel Storage (TESTSP)

Utility subprograms are documented in section 19.

The opening paragraph for each processor describes briefly what the processor does and its subprogram structure, including utilities. A linkage diagram illustrates the detailed structure, with subprograms listed alphabetically. Each subprogram is described in a succeeding subsection.

The description of each subprogram begins with an initial, concise statement of the routine's function or functions. Then, more detailed information is provided on the following:

- Linkage — the subprograms called and those which call the routine.
- Interfaces — common blocks and calling arguments, if applicable.
- Inputs — tape files from other processors, if input directly to the subprogram; calling sequence, if applicable, along with a list of parameters, their dimensions, definitions, and whether input or output; and card images such as control cards and class, subclass, and field definition cards.
- Storage requirements — stated in bytes.
- Description — a more detailed description of the subprogram functions, if necessary. (If adequately stated in the opening paragraph, this description is omitted.)
- Flow chart — if available, flow charts are placed (alphabetically) in the final subsection after all subprograms have been documented.
- Listing — published in volume IV of this user guide.

This document is intended as a user aid in interpreting the structure of the EOD-LARSYS and as an informational source as to the functions of the various subprograms. It does not define or describe the in-depth mathematical and statistical procedures that are performed during the execution of each processor. The user is referred to volume II for available information of this type and for diagnostic messages, file and card image formats, and other specific information required to execute the system.

Additional programs which are executable on the EOD-LARSYS and which will be documented separately are: the clustering algorithm CLASSY;* the Texas A&M classification program AMOEBA;† and the Equi-Probable Blocks (EQUPRB),† Multitemporal Bayes (MULBAY),† and Principal Component Greenness (PCG)* programs.

*See section 2.

†No documentation is available at this time.

TABLE 1-1.- ALPHABETICAL INDEX OF EOD-LARSYS SUBPROGRAMS

Subprogram	Processor	Function	Section
ADDRES	NDHIST	Allocates storage space and computes addresses.	15.2
ALLKIN	LABEL	Labels statistics using all-of-a-kind procedure.	18.2
ALPHA	GTDDM	Checks for alphabetic character and returns integer value of the character.	21.2
AMFIL	TRSTAT	Uses formatted read to retrieve the A-transformation matrix (A-matrix) and B-vector card image files.	14.2
AMFILE	TRSTAT	Uses unformatted read to retrieve the A-matrix and B-vector card image files.	14.3
ASCEND	LABEL	Sorts floating-point numbers in ascending order.	18.3
AVEDIV	SELECT	Computes weighted average interclass divergence and partial derivatives with respect to the B-transformation matrix (B-matrix).	10.2
BHTCHR	SELECT	Computes interclass Bhattacharyya distance, weighted average interclass divergence, and partial derivatives with respect to the B-matrix.	10.3
BMFIL	Utility	Performs input/output functions with respect to the B-matrix.	19.1
BNI4A1	Utility	Converts internal binary to EBCDIC characters.	19.2
BSTCHK	SELECT	Checks the validity of user-requested channels (features);[a] entry EVLCHK generates a corrected channel request queue.	10.4
BUFILL	Utility	Reads the multispectral scanner (MSS) image data tape (DATAPE) one record at a time.	19.3
CATGRY	CLASSIFY	Category classifier.	11.2
CATSCN	CLASSIFY	Reads CATEGORY control card and stores category and class names.	11.3
CHAIN	Utility	Links all clusters (subclasses)[a] having means less than a specified distance apart.	19.4
CHI	DISPLAY	Calculates chi-squared distribution with n degrees of freedom.	12.2

[a]The words feature and channel and cluster and subclass are synonymous and are used interchangeably in this document.

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
CHIN	DISPLAY	Calculates chi-squared threshold values.	12.3
CHLDET	Utility	Calculates modified Cholesky decomposition and determinant of the covariance matrix.	19.5
CLDIST	Utility	Calculates weighted distance between cluster means.	19.6
CLOCK	-	Records execution time for each processor; called by MONTOR and MONPAC.	(b)
CLRCOD	SCTRPL	Retrieves values for color codes to be output on tape.	16.2
CLRKEY	LABEL	Writes the color keys on the mixed and/or conditional cluster map (MAPUNT) file.	18.4
CLRKYS	SCTRPL	Adds the color keys to color-coded spectral plot image tapes.	16.3
CLSCHK	Utility	Checks user-input classes, subclasses, groupings, and channels.	19.7
CLSCOV	STAT	Entry to FLDCOV.	8.3
CLSFY	CLASSIFY	Driver routine.	11.1
CLSFY1	CLASSIFY	Performs subclass classification.	11.4
CLSFY2	CLASSIFY	Classifies by grouping subclasses into fields or categories.	11.5
CLSHIS	Utility	Scales and prints a histogram of training subclass and field data.	19.8
CLSMAP	LABEL	Outputs conditional and/or mixed MAPUNT file.	18.5
CLSSPC	STAT	Generates a spectral plot for each user-defined training field and/or subclass; entry FLDSPC plots spectral responses for training fields, and entry MULSPC plots multispectral responses for all training subclasses.	8.2
CMERR	Utility	Prints an error message and terminates execution.	19.9
CNDMAP	LABEL	Flags conditional clusters.	18.6
CNTER	SCTRPL	Retrieves frequency count for histogrammed vector of interest.	16.4

[b]Not documented.

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
COLINV	SELECT	Inverts a given symmetric positive definite matrix S.	10.5
COMHST	Utility	Entry to CLSHIS.	19.8
CONTEX	CLASSIFY	Performs standard or maximum likelihood classification.	11.6
CONVRT	SELECT	Converts EBCDIC characters to computational integers or computational integers to EBCDIC characters.	10.6
COVAR1	ISOCLS	Calculates and prints the covariance matrix for each cluster.	9.2
COVPAT	TESTSP	Calculates and prints the covariance matrix for each cluster.	23.2
CRDSCN	LABEL	Interprets the control card images DOTLABEL and STATLABEL.	18.7
CRDSTA	Utility	Reads card image input, computes base addresses, and stores data in ARRAY.[c]	19.10
CUBIC	SELECT	Generates the cubic fit technique of minimizing a function using the Davidon-Fletcher-Powell method.	10.7
DAMRG	DAMRG	Driver routine.	20.1
DATATR	DATA-TR	Driver routine.	13.1
DAVDN1	SELECT	Initializes the H and P arrays used in the Davidon-Fletcher-Powell method.	10.8
DAVDN2	SELECT	Performs a one-dimensional search for the best k of n channels to use in executing the Davidon-Fletcher-Powell procedure.	10.9
DAVDN3	SELECT	Updates the H and P arrays for the next cycle of the search for the best k of n channels to use in executing the Davidon-Fletcher-Powell procedure.	10.10
DAVIDN	SELECT	Driver routine for the Davidon-Fletcher-Powell procedure.	10.11
DDM	GTDDM	Processes and labels 209 dots.	21.3
DESCEN	Utility	Arranges a set of integers in descending order.	19.11

[c]ARRAY is a block of working storage passed to each processor for the variable dimensioning of other arrays.

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
DESIG	DISPLAY	Sets the IR output array for DO/DU[d] fields and locates DO/DU fields on each line.	12.4
DISTCV	DISPLAY	Plots distribution and chi-squared curves and computes empirical threshold values.	12.5
DIVERG	SELECT	Computes average interclass divergence; entry DIVRG1 computes the same using the double-precision subclass covariance matrix and mean vector.	10.12
DIVRG1	SELECT	Entry to DIVERG.	10.12
DOTDAT	DOTDATA	Driver routine.	17.1
DOTDST	LABEL	Computes and stores intercluster dot distances.	18.8
DOTS	DOTDATA	Coordinates and supervises functions to output a dot data file on the dot unit (DOTUNT).	17.2
DSPLAY	DISPLAY	Driver routine.	12.1
DSPLY1	DISPLAY	Reads, writes, and stores statistics from the classification map tape (MAPTAP) file.	12.6
DSPLY2	DISPLAY	Multifunction routine for displaying classification results.	12.7
DSPTAP	LABEL	Writes an unformatted, conditional cluster map file on MAPUNT.	18.9
DSTAPE	Utility	Generates a formatted cluster map file on MAPUNT.	19.12
DWRTMX	Utility	Entry to WRTMTX.	19.59
EMTHRS	DISPLAY	Calculates empirical thresholds and plots histograms of the quadratic form.	12.8
EVALSP	SELECT	Coordinates the routines for computing the separability measure for a particular set of features.	10.13
EVLCHK	SELECT	Entry to BSTCHK.	10.4
EVLFET	SELECT	Evaluates user-input channels for selection of the best k of n channels.	10.14
EXSRCH	SELECT	Uses the Exhaustive Search procedure to select the best k of n channels.	10.15

[d]Designated other/designated unidentifiable.

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
FALSY	CLASSIFY	Finds a root of the polynomial G used in computing class-pair thresholds.	11.7
FDIST	DISPLAY	Coordinates the function used to obtain the Fisher F-distribution threshold values.	12.9
FDLINT	Utility	Accepts a nonrectangular field table as input and returns the x-intercepts for the given scan line.	19.13
FILERD	LABEL	Computes high-speed disk addresses and reads input files.	18.10
FIND12	Utility	Scans card image input and locates special symbols.	19.14
FINT1	SELECT	Initializes addresses and prints header; entry FINT2 evaluates the separability measure and computes partial derivatives for a given channel.	10.16
FINT2	SELECT	Entry to FINT1.	10.16
FISH	DISPLAY	Calculates Fisher F-distribution threshold values.	12.10
FISHIN	DISPLAY	Used in calculating the Fisher F-distribution threshold values.	12.11
FLDBOR	DISPLAY	Sets symbol index for outlining training or test fields on the classification map.	12.12
FLDCLS	NDHIST	Supervises reading of field definition data set when data are to be grouped by class.	15.3
FLDCOV	STAT	Calculates covariance matrices and correlation coefficients for training fields; entry CLSCOV calculates the same statistics for training subclasses.	8.3
FLDFLD	NDHIST	Supervises reading of field definition data set when data are to be grouped by field.	15.4
FLDHIS	Utility	Entry to CLSHIS.	19.8
FLDINT	Utility	Unpacks picture elements (pixels) from specified lines on the MSS DATAPE.	19.15
FLDLAC	DOTDATA	Reads the DOTUNT file and field definition card images, computes line and sample increments, and stores dot information.	17.3
FLDMEN	NDHIST	Computes mean of each requested channel for each input test and/or training field.	15.5

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
FLDSPC	STAT	Entry to CLSSPC.	8.2
FLDSUB	NDHIST	Supervises reading of field definition data set when data are to be grouped by subclass.	15.6
FLDTYP	DOTDATA	Initiates reading of field definition data set and signals the calling routine when all fields of a given type have been processed.	17.4
FLTNUM	Utility	Scans card images, interprets real numbers separated by commas, and returns them in an array.	19.16
FSBSFL	Utility	Positions file for an unformatted read or write.	19.17
FSFMFL	Utility	Positions file for a formatted read or write.	19.18
G	CLASSIFY	Polynomial from which the class-pair threshold is computed.	11.8
GENRPT	SELECT	Generates results of channel selection on the line printer.	10.17
GETINF	Utility	Retrieves stored information.	19.19
GETSET	SELECT	Produces a unique set of channels from an input channel set.	10.18
GETST	Utility	Retrieves, formats, writes, and stores data from the statistics tape (SAVTAP) file.	19.20
GJR	CLASSIFY	Used to calculate the polynomial G.	11.9
GRAYMP	GRAYMAP	Driver routine.	7.1
GRPSCN	Utility	Scans field definition card images and determines if classes or subclasses have been assigned to groups.	19.21
GTCRPL	GTTCN	Examines subpixel labels and labels the pixel.	22.2
GTDDM	GTDDM	Driver routine.	21.1
GTDOTS	GTDDM	Extracts and stores crop code values.	21.4
GTDTL	GTDDM	Performs a data transformation on crop codes and builds a table of existing categories.	21.5
GTDWR	GTDDM	Outputs LACIE-formatted dot files.	21.6
GTSTAT	SELECT	Retrieves the reduced channel covariance matrices and subclass mean vectors.	10.19

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
GTTCN	GTTCN	Driver routine.	22.1
GTTRNS	GTDDM	Constructs the default transformation of crop code numbers to one-character crop names.	21.7
GTUNPK	GTTCN	Converts the crop code to a numerical designation.	22.3
HEADNG	GRAYMAP	Prints heading for the pictorial display of histogrammed data.	7.2
HIST	HIST	Driver routine.	6.1
HISTGM	Utility	Calculates histograms and writes histogrammed statistics on file.	19.22
HISTIC	Utility	Computes and displays statistics for the histogram.	19.23
HSTGRM	Utility	Entry to CLSHIS.	19.8
I4A1BN	Utility	Converts EBCDIC digits to internal binary integers.	19.24
ISOCLS	ISOCLS	Driver routine.	9.1
ISODAT	ISOCLS	Coordinates the clustering process.	9.3
ISOPAT	TESTSP	Coordinates the clustering process.	23.3
KBTRAN	DATA-TR	Rescales transformed data using the statistical method.	13.2
KNEAR	LABEL	Labels statistics using the k-nearest-neighbor procedure.	18.11
LABDOT	LABEL	Labels or relabels the DOTUNT file.	18.12
LABEL	LABEL	Driver routine.	18.1
LABLR	LABEL	Coordinator and supervisor of routines required to perform analyst-specified operations.	18.13
LABMAN	Utility	Writes a formatted SAVTAP file and creates a module statistics (module STAT) file.	19.25
LAREAD	Utility	Reads field definition card images.	19.26
LEARN	STAT	Computes statistics for user-defined training fields and/or subclasses.	8.4
LINERD	Utility	Reads one scan line of data from the MSS DATAPE.	19.27

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
LINLAB	GTTCN	Converts three lines of input data to one line of output data.	22.4
LINPLT	SCTRPL	Constructs pixel frequency plot on disk and prints it on the line printer.	16.5
LISTLC	Utility	Reads dot data files, computes line and sample increments, and stores data.	19.28
LISTPR	DISPLAY	Prints three classification performance tables of dot data.	12.13
LISTSM	DISPLAY	Supports Label Identification From Statistical Tabulation (LIST) processing.	12.14
LNTRAN	DATA-TR	Initiates data transformation, rescales and histograms data, and supervises distribution of transformed data.	13.3
MANORD	LABEL	Relabels the SAVTAP file.	18.14
MAPHD	DISPLAY	Prints header information for the MAPTAP file.	12.15
MAPHDG	CLASSIFY	Prints header for the MAPTAP file.	11.10
MAPHND	LABEL	Prints heading for the conditional and/or mixed MAPUNT file.	18.15
MATTNS	SCTRPL	Multiplies a matrix A by a vector B to obtain a one-dimensional matrix C; also adds a vector D to C.	16.6
MATVEC	Utility	Multiplies a matrix A by a vector B and stores the result in a vector C.	19.29
MAXMAT	DATA-TR	Computes approximate transformed maximum and minimum for each component in the data transformation.	13.4
MCHLSK	CLASSIFY	Computes the modified Cholesky decomposition of the covariance matrix.	11.11
MIXMAP	LABEL	Flags conditional clusters.	18.16
MONPAC	-	EOD-LARSYS system monitor.	3
MONTOR	-	EOD-LARSYS system monitor.	3
MSCAN	-	Reads and analyzes control card images and supervises MONTOR to call the various processors.	4

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
MTMDAT	Utility	Multiplies a matrix A by the transpose of a matrix B and stores the result in a lower triangular matrix DD.	19.30
MTMLS6	Utility	Multiplies a matrix A by a vector B and stores the result in matrix C.	19.31
MT1	SELECT	Stores a double-precision matrix A in array B in symmetric notation and single precision.	10.20
MT2	SELECT	Multiplies double-precision matrices A and B and stores the product in array C in symmetric notation and double precision.	10.21
MT3	SELECT	Multiplies and stores two double-precision matrices, one or both in symmetric notation.	10.22
MT4	SELECT	Stores a matrix A either in full or symmetric notation and in double precision.	10.23
MULSPC	STAT	Entry to CLSSPC.	8.2
NAMSTA	Utility	Provides cluster names based on category names and sequence numbering.	19.32
NDHIST	NDHIST	Driver routine.	15.1
NDHST1	NDHIST	Coordinates the histogramming process.	15.7
NDHST2	NDHIST	Computes a 1- to 16-channel histogram for either one or two sets of data.	15.8
NUMBER	Utility	Scans control card images searching for integers.	19.33
NUMBR	Utility	Processes one field definition card image at a time; reads and stores all numbers in array NDOTS, with ND CARD as an index.	19.34
NXTCHR	Utility	Locates and enters the next nonblank character on a card image being read.	19.35
OFFSET	SCTRPL	Computes the value of each sample (x-axis) and line (y-axis) location on a scan line.	16.7
ORDER	Utility	Arranges a set of N integers in ascending order.	19.36
PCT	DISPLAY	Builds the classification performance table or plots the histogram of the quadratic form for empirical thresholds (entry PCTT).	12.16
PCTT	DISPLAY	Entry to PCT.	12.16

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
PICOLR	NDHIST	Extracts radiance values from unpacked data for use as color codes on the scatter plot tape (SCTRUN).	15.9
PICT	GRAYMAP	Pictorially displays the requested features.	7.3
PLOT	SELECT	Generates a plot of the results of channel selection.	10.24
PRELIM	SELECT	Performs some of the preliminary tasks for feature selection.	10.25
PRINT	Utility	Generates most of the output for the ISOCLS and TESTSP processors.	19.37
PRTCOV	Utility	Writes the transformed covariance matrix.	19.38
PRTFLD	SELECT	Coordinates the printing of training fields and subclass statistics.	10.26
PRTPCT	DISPLAY	Prints the classification performance table.	12.17
PRTPLT	SCTRPL	Entry to LINPLT.	16.5
PRTSUM	DISPLAY	Prints the classification performance summary and/or intensive test site (ITS) summary reports.	12.18
PSPLIT	ISOCLS	Assigns pixels to clusters and computes statistics.	9.4
PSPPAT	TESTSP	Assigns pixels to clusters and computes statistics.	23.4
RANK	Utility	Calculates greenness value for each cluster and outputs an ordered list of color keys, cluster numbers, and greenness values.	19.39
RDDATA	ISOCLS	Coordinates the routines which read fields of data from the MSS DATAPE and store the data on disk for processing.	9.5
RDDOTS	Utility	Reads the DOTUNT file.	19.40
RDDOT1	Utility	Retrieves spectral and/or spatial information from the DOTUNT file.	19.41
RDDPAT	TESTSP	Coordinates the routines which read fields of data from the MSS DATAPE and store the data on disk for processing.	23.5
RDFILE	Utility	Entry to RDMEAN.	19.42

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
RDMEAN	Utility	Reads the MEAN card image file; entry RDFILE reads the file, stores requested channels, and prints a summary of means for requested channels.	19.42
RDMODK	Utility	Reads in remainder of the module STAT file and writes statistics on the SAVTAP file.	19.43
REDDAT	Utility	Reads covariances and means from the SAVTAP file and reduces statistics.	19.44
REDIF2	CLASSIFY	Reads and analyzes supervisor control card images.	11.12
REDIF3	DISPLAY	Reads and analyzes supervisor control card images.	12.19
REDSAV	Utility	Reads the SAVTAP file and reduces statistics to a user-requested subset.	19.45
RELERR	CLASSIFY	Computes the Euclidean norm of the covariance matrix before Cholesky factorization.	11.13
REODER	LABEL	Reorders subclass names and numbers of associated pixels as part of the manual relabeling of statistics option.	18.17
RESCLE	SCTRPL	Rescales transformed data to a user-specified range.	16.8
RESTO	NDHIST	Reads a scan line of data from MAPUNT into core; entry RESTOR returns K, pixel of interest, to the calling routine.	15.10
RESTOR	NDHIST	Entry to RESTO.	15.10
RNORM	DISPLAY	Calculates the chi-square for 1 degree of freedom.	12.20
RREAD	Utility	Simulates the random read of a work file used to store data temporarily during execution.	19.46
RWRITE	Utility	Simulates the random write of a work file used to store data temporarily during execution.	19.47
SAVFIL	Utility	Reads records from the SAVTAP file.	19.48
SCALE	SELECT	Reduces x- and y-coordinates for plotting channel selection results.	10.27
SCATTR	SCTRPL	Sets up the logic and structure for creating color-coded spectral plots.	16.9

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
SCTRPL	SCTRPL	Driver routine.	16.1
SEARCH	Utility	Searches for the correct scan line to process.	19.49
SELECT	SELECT	Driver routine.	10.1
SETADR	SCTRPL	Computes addresses for ARRAY.	16.10
SETMRG	Utility	Inactive.	19.50
SETREM	DATA-TR	Unpacks input scale parameters, checks to assure one-to-one correspondence with additive scaling bias, and stores values.	13.5
SETUP1	STAT	Reads and analyzes control card images and sets options.	8.5
SETUP2	CLASSIFY	Analyzes supervisory information and reduces statistics for processing.	11.14
SETUP3	DISPLAY	Reads and analyzes control card images, locates files on the MAPTAP and DOTUNT, and sets options.	12.21
SETUP4	SELECT	Reads and analyzes control card images and sets options.	10.28
SETUP5	HIST	Reads and analyzes control card images and sets options.	6.2
SETUP6	GRAYMAP	Reads and analyzes control card images and sets options.	7.4
SETUP7	Utility	Reads and analyzes control card images and sets options.	19.51
SETUP8	DATA-TR	Reads and analyzes control card images and sets options.	13.6
SETUP9	TRSTAT	Reads and analyzes control card images and sets options.	14.4
SETUS	CLASSIFY	Uses the lower triangular matrix and the diagonal matrix for a specific class to compute the full covariance matrix; also extracts the mean vector for the class.	11.15
SET10	NDHIST	Reads and analyzes control card images and sets options.	15.11
SET11	SCTRPL	Reads and analyzes control card images and sets options.	16.11

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
SET13	DOTDATA	Reads and analyzes control card images and sets options.	17.5
SET14	LABEL	Reads and analyzes control card images and sets options.	18.18
SET17	GTTCN	Analyzes control card images and sets options.	22.5
SET18	DAMRG	Reads and analyzes control and field definition card images and sets options.	20.2
SET19	GTDDM	Reads and analyzes control card images and sets options.	21.8
SORT	HIST	Arranges a vector of integers in ascending order.	6.3
SORTVC	SCTRPL	Sorts the elements of a floating-point array in ascending order.	16.12
STAT	STAT	Driver routine.	8.1
STODAT	NDHIST	Reads the MAPUNT file and stores the image on high-speed disk.	15.12
STOFIL	SCTRPL	Reads the remainder of the N-dimensional histogram tape (NHSTUN) file and stores data on high-speed disk.	16.13
STOMAP	LABEL	Reads the MAPUNT file and stores the image on high-speed disk.	18.19
STOPTS	SCTRPL	Entry to LINPLT.	16.5
SUNFAC	Utility	Computes Sun-angle gain corrections for pixel radiance values for each channel based on input Sun angles.	19.52
TAPHDR	Utility	Reads the header record of the MSS DATAPE (formatted read).	19.53
TCN	GTTCN	Converts Universal-formatted 351-by-392 (sample-by-line) ground-truth image files to Universal-formatted 117-by-196 files.	22.6
TESTSP	TESTSP	Driver routine.	23.1
THRESH	CLASSIFY	Computes class-pair thresholds.	11.16
TINORM	DISPLAY	Calculates an approximation to inverse normal distribution for the chi-squared and Fisher F-distribution threshold values.	12.22

TABLE 1-1.- Continued.

Subprogram	Processor	Function	Section
TNSFER	SCTRPL	Coordinates the reduction of each 1- to 16-element data vector taken from the NHSTUN file to two components.	16.14
TRACE	SELECT	Computes the trace of the product of two symmetric matrices and stores the result in symmetric notation and double precision.	10.29
TRAMTX	TRSTAT	Performs statistical transformation and outputs results to the SAVTAP file or card images (optional).	14.5
TRANSF	DATA-TR	Performs a linear data transformation.	13.7
TRHIST	DATA-TR	Computes scaling parameters for transformed data using a histogram of the transformed data.	13.8
TRNDIV	SELECT	Computes average weighted transformed divergence and partial derivatives with respect to the B-matrix.	10.30
TRNSFR	SELECT	Multiplies the B-matrix times its transpose and times a matrix A and stores the result in double precision.	10.31
TRSTAT	TRSTAT	Driver routine.	14.1
UNPCKV	SCTRPL	Unpacks and stores a two-channel vector from the NHSTUN file.	16.15
USERIN	SELECT	Coordinates the routines required to compute the separability measure for the user-input B-matrix.	10.32
VECSCN	SCTRPL	Scans the COLOR control card, changes alphanumeric characters to integer mode, and returns one character at a time to the calling routine.	16.16
WGTCHK	SELECT	Sets up a table of intersubclass weights.	10.33
WGTSCN	SELECT	Scans the WEIGHTS control card and saves class name pairs and associated weights.	10.34
WHRPLC	SELECT	Coordinates the routines for finding the best k of n channels using the Without Replacement procedure.	10.35
WRTAMT	TRSTAT	Outputs transformed statistics on the printer.	14.6
WRTBMT	Utility	Writes the transformed B-matrix.	19.54

TABLE 1-1.- Concluded.

Subprogram	Processor	Function	Section
WRTDOT	Utility	Outputs the DOTUNT file for the LABEL and DOTDATA processors.	19.55
WRTFIL	NDHIST	Writes the NHSTUN file.	15.13
WRTFLD	Utility	Prints saved training or test fields.	19.56
WRTHED	Utility	Writes the header record for each data tape (formatted write).	19.57
WRTLN	Utility	Writes the data for each data tape (formatted write).	19.58
WRTMTX	Utility	Prints the single-precision covariance matrices; entry DWRTMX prints the double-precision covariance matrix.	19.59
WRTREC	Utility	Prints a scan line (one record) of data.	19.60
Special routine			
BLKCOM	All processors	Initializes variables in the GLOBAL common block by means of data statements.	(e)

[e]Not documented.

2. APPLICABLE DOCUMENTS

Information was extracted from, or reference was made to, the following documents in the preparation of this volume III.

1. Stewart, J.; et al.: EOD-LARSYS User Guide for the IBM 370/148 — vol. I, System Overview. JSC-13821, LEC-12563, NASA/JSC (Houston), Aug. 1978.
2. Stewart, J.; et al.: EOD-LARSYS User Guide for the IBM 370/148 - vol. II, User's Reference Manual. JSC-13821, LEC-12564, NASA/JSC (Houston), Dec. 1978.
3. Burnell, M. L.; and Aucoin, P. J.: EOD-LARSYS User Guide for the IBM 370/148 — vol. IV, Program Listings. JSC-13821, LEC-12566, rev. A, NASA/JSC (Houston), Nov. 1979.
4. Minter, R. T.; et al.: User Documentation, EOD-LARSYS. JSC-12504, LEC-3984, rev. 4, NASA/JSC (Houston), July 1977.
5. Wills, B. E.; et al.: "As-Built" Design Specification for EOD-LARSYS Procedure 1. JSC-13143, LEC-11293, NASA/JSC (Houston), Oct. 1977.
6. Aucoin, P. J.: Final Design Specification for EOD-LARSYS Procedure 1 Follow-on. JSC-13817, LEC-11618, NASA/JSC (Houston), Dec. 1977.
7. Wills, B. E.: Final Design Specification for Color-Coded Spectral Plots Program. LEC-10068, NASA/JSC (Houston), Apr. 1977.
8. Rowland, J. K.: Final Design Specification for EOD-LARSYS/ Data Transformation Processor Modification. JSC-12917, LEC-10662, NASA/JSC (Houston), Apr. 1977.
9. Minter, R. T.: Computer Program Documentation, ISOCLS. MSC Program Number C094, NASA/JSC (Houston), Oct. 1972.
10. Aucoin, P. J.; and Gor, J.: "As-Built" Design Specification for LACIE-Formatted Dot Cards in EOD-LARSYS. JSC-13972, LEC-12154, NASA/JSC (Houston), Apr. 1978.

11. Horton, C. L.; and Lennington, R. K.: "As-Built" Design Specification for CLASSY and Adaptive Maximum Likelihood Clustering Method. JSC-16466, LEMSCO-14546, NASA/JSC (Houston), Feb. 1980.

12. Welter, D. E.: "As-Built" Design Specification for Principal Component Greenness (PCG) Transformation Processor as Implemented on the Earth Observations Division Version of the Laboratory for Applications of Remote Sensing System (EOD-LARSYS). LEMSCO-14279, NASA/JSC (Houston), to be published.

3. MONTOR AND MONPAC

The MONTOR and MONPAC executive routines are the EOD-LARSYS system monitors for the various processors. MONPAC is used when the TESTSP processor is involved in place of the ISOCLS processor. The functions of MONTOR and MONPAC are the same, the storage allocation being the basic difference between the two routines.

3.1 LINKAGES

The executive routine calls the CLOCK, CLSFY, DAMRG, DATATR, DOTDAT, DSPLAY, GRAYMP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, MSCAN, NDHIST, SCTRPL, SELECT, STAT, and TRSTAT subprograms. It is the initial routine and is not called by any other routine.

3.2 INTERFACES

The executive routine interfaces with other routines through common blocks ARRAY and GLOBAL and through the calling arguments.

3.3 INPUTS

Input to the executive routine consists of a user-specified tape file and processor control cards.

3.4 OUTPUTS

This program outputs the execution time for each processor.

3.5 STORAGE REQUIREMENTS

The executive routine requires 2710 bytes of storage.

3.6 DESCRIPTION

The executive routine allocates storage for ARRAY; calls a block common routine to initialize parameters for the GLOBAL common block (section 5), which is used in every processor; and invokes system routine RINIT to assign the random access disk file for scratch (used by the DISPLAY, GRAYMAP, ISOCLS, and SELECT processors). It then calls MSCAN (section 4), which directs calls to the various processors according to the value of the input parameter JGO.

3.7 FLOW CHART

Figure 3-1 is a functional flow diagram of the executive routine.

3.8 LISTING

The listings for MONTOR and MONPAC are provided in volume IV, section 3, of this user guide.

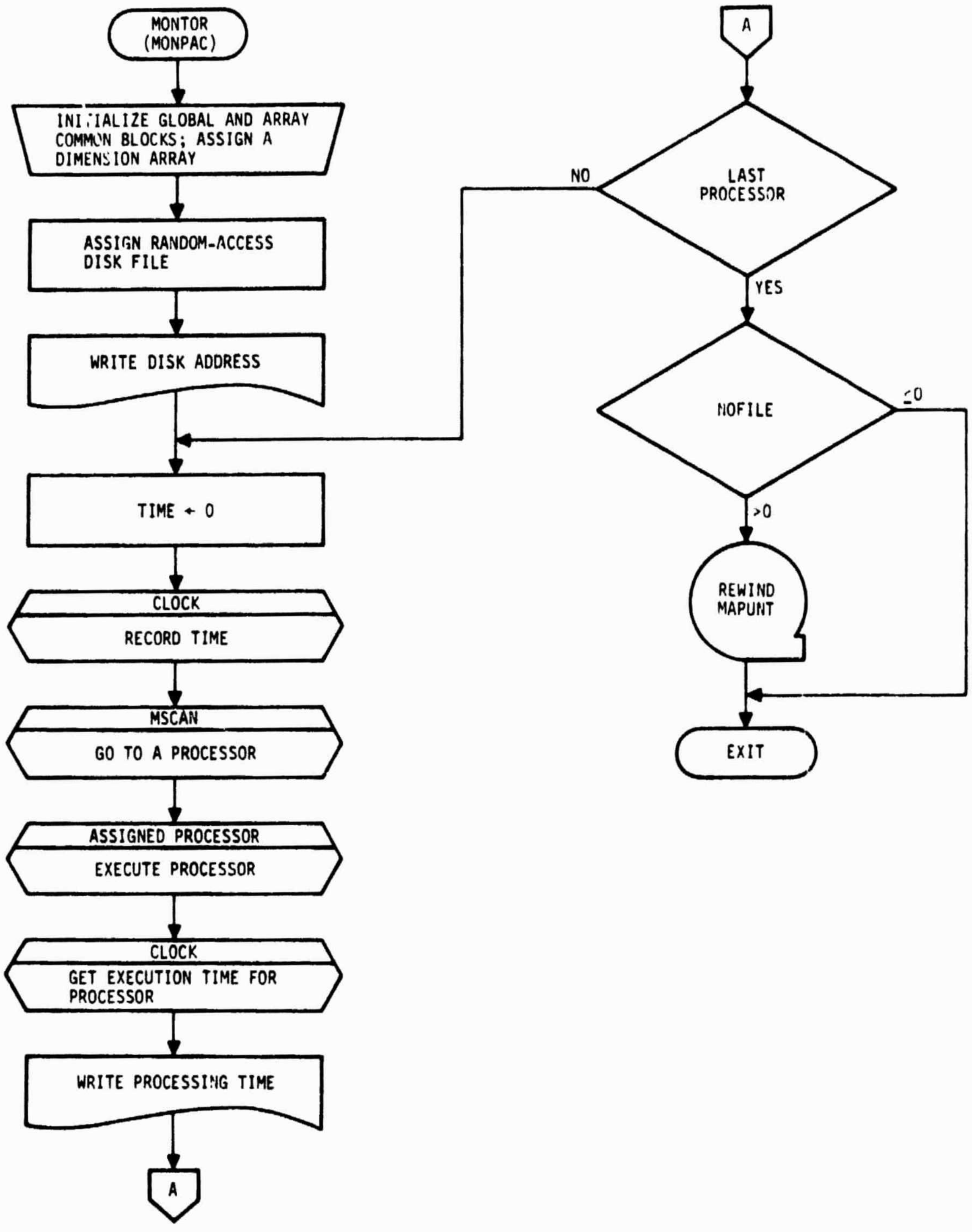


Figure 3-1.- Functional flow chart for the executive routine.

4. MSCAN

The MSCAN subprogram reads and analyzes all processor control card images and directs the MONTOR or MONPAC executive routine to call the various system processors.

4.1 LINKAGES

This routine calls the NUMBER and NXTCHR subprograms. It is called by the executive routine.

4.2 INTERFACES

The MSCAN subprogram interfaces with other routines through common blocks GLOBAL, IDSTOR, and TAPERD and through the calling arguments.

4.3 INPUTS

Calling sequence: CALL MSCAN(JGO,DBUG)

Parameter	Dimension	In/out	Definition	
JGO	1	In	Processor pointer:	
			1 = $STAT	13 = $DOTDAT
			2 = $CLASFY	14 = $LABEL
			3 = $DSPLAY	15 = $EQUPRB
			4 = $SELECT	16 = $MULBAY
			5 = $HIST	17 = $GTTCN
			6 = $ISOCLS	18 = $DAMPG
			7 = $GRAYMP	19 = $AMOEBA
			8 = $DATATR	20 = $CLASY
			9 = $SIGEXT*	21 = $TESTSP
			10 = $TRSTAT	22 = $GTDDM
			11 = $NDHIST	23 = $PCG
			12 = $SCTRPL	

*No longer in use.

Parameter	Dimension	In/out	Definition
DBUG	1	In/out	If = 1, first entry to MSCAN.

The control cards relevant to this routine are given in section 3.2.2 of volume II of this user guide.

4.4 OUTPUTS

This subprogram outputs the date and standard heading on the printer and lists the contents of input monitor control cards.

4.5 STORAGE REQUIREMENTS

The MSCAN subprogram requires 1782 bytes of storage.

4.6 DESCRIPTION

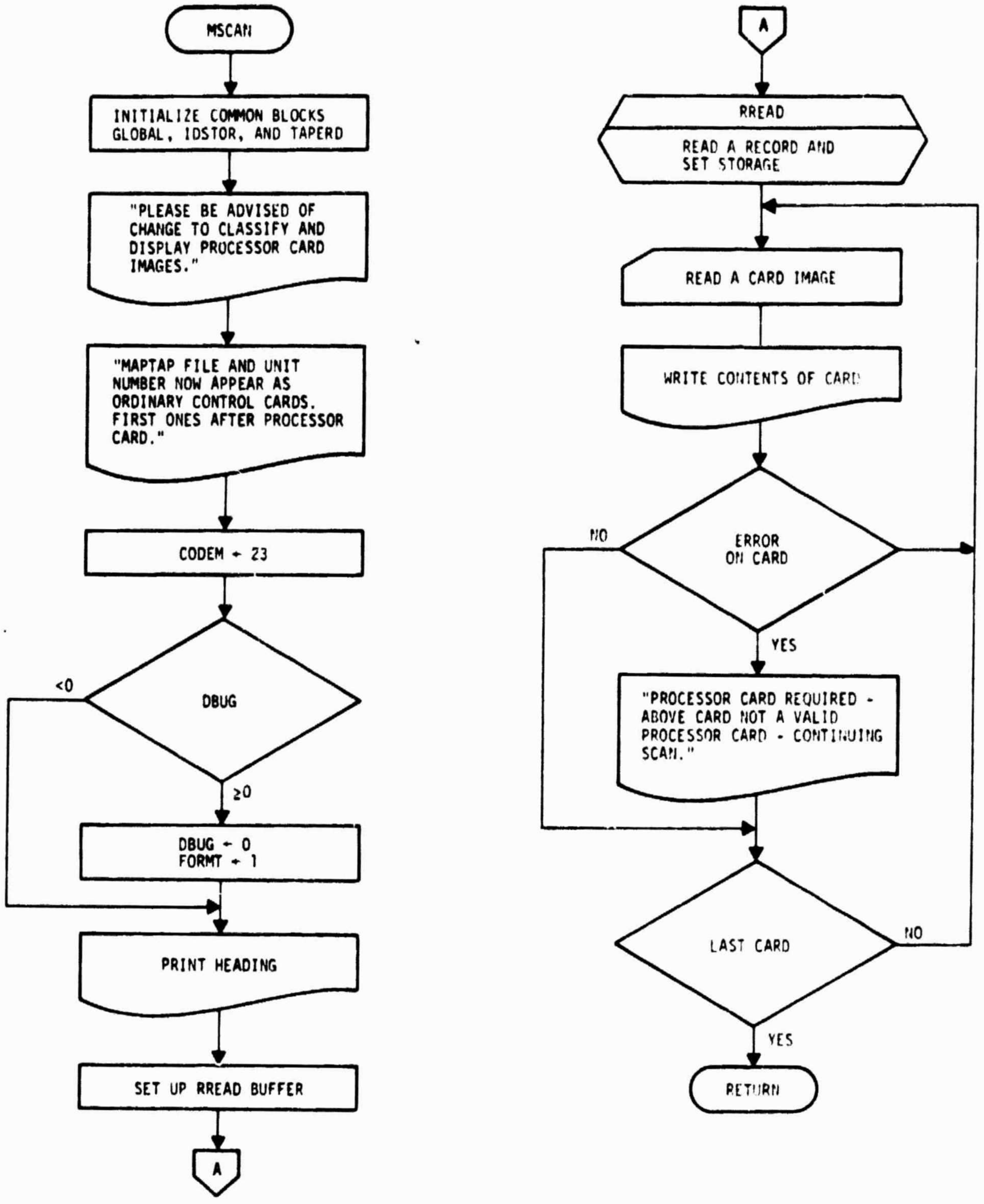
The MSCAN subprogram accepts processor control cards as input. It sets the date of execution, writes the heading at the top of the printed page, reads and decodes the monitor card, and sets up the RREAD buffer to read card images. It generates error messages if a processor card is missing or in error.

4.7 FLOW CHART

Figure 4-1 is a functional flow diagram of the MSCAN subprogram.

4.8 LISTING

The subprogram listing is given in volume IV, section 4, of this user guide.



MSCAN
INITIALIZE COMMON BLOCKS GLOBAL, IDSTOR, AND TAPERD
"PLEASE BE ADVISED OF CHANGE TO CLASSIFY AND DISPLAY PROCESSOR CARD IMAGES."
"MAPTAP FILE AND UNIT NUMBER NOW APPEAR AS ORDINARY CONTROL CARDS. FIRST ONES AFTER PROCESSOR CARD."
CODEM ← 23
DBUG
<0
≥0
DBUG ← 0
FORMT ← 1
PRINT HEADING
SET UP RREAD BUFFER
A
A
RREAD
READ A RECORD AND SET STORAGE
READ A CARD IMAGE
WRITE CONTENTS OF CARD
ERROR ON CARD
NO
YES
"PROCESSOR CARD REQUIRED - ABOVE CARD NOT A VALID PROCESSOR CARD - CONTINUING SCAN."
LAST CARD
NO
YES
RETURN

Figure 4-1.- Functional flow chart for the MSCAN subprogram.

5. COMMON BLOCKS AND BLOCK DATA

The common blocks used within the EOD-LARSYS, along with their processor and subprogram interfaces, are listed in table 5-1. Parameter definitions for and a brief description of each common block are given in the following subsections.

TABLE 5-1.- EOD-LARSYS COMMON BLOCKS AND INTERFACES

Common block	Interfaces	
	Processor	Subprogram
ARRAY	All processors	MONITOR and MONPAC
BESTKN	SELECT	PRELIM, REDDAT, SELECT, and SETUP4
BMTRX	CLASSIFY	CLSFY
BUFF	CLASSIFY, DAMRG, DATA-TR, DOTDATA, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, STAT, and TESTSP	LINERD
CLASS	CLASSIFY	CATGRY, CLSFY, CLSFY1, CLSFY2, CONTEX, REDIF2, and SETUP2
DISPL	DISPLAY	DISTCV, DSPLAY, DSPLY1, DSPLY2, EMTHRS, FDIST, PRTPCT, PRTSUM, REDIF3, and SETUP3
DOTVEC	DISPLAY and DOTDATA	LISTLC (utility subprogram); DOTS, FLDLAC, FLDTYP, LISTSM, and SET13
DVNBLK	SELECT	DAVDN1, DAVDN2, DAVDN3, and DAVIDN
FNTDUM	SELECT	FINT1
FSL	SELECT	AVEDIV, BHTCHR, BSTCHK, DAVDN1, DAVDN2, DAVIDN, EVALSP, EVLFET, EXSRCH, FINT1, GENRPT, GTSTAT, PLOT, PRELIM, PRTFLD, SCALE, SELECT, SETUP4, TRNDIV, TRNSFR, USERIN, and WHRPLC
GLOBAL	All processors	BMFIL, CHAIN, CLSHIS, CRDSTA, DSTAPE, HISTGM, HISTIC, MSCAN, PRINT, PRTCOV, RANK, RDDOTS, RDMEAN, RDMODK, REDDAT, REDSAV, SAVFIL, SETUP7, SUNFAC, WRTBMT, and WRTFLD (utility subprograms); ALLKIN, AMFIL, AMFILE, CLSFY, CLSFY1, CLSFY2, CLSMAP, CLSSPC, COVAR1, COVPAT, DAMRG, DATATR, DOTDST, DOTS, DSPLY1, DSPLY2, DSPT[illegible], EMTHRS, FILERD, FINT1, FLDCOV, GENRPT, GRAYMP, HEADNG, ISOCLS, ISODAT, ISOPAT, KBTRAN, KNEAR, LA[illegible]R, LEARN, LINPLT, LNTRAN, MIXMAP, NDHST1, NDHST2, OFFSET, PICT, PLOT, PRTFLD, PRTPCT, PRTSUM, RDDATA, RDDPAT, REDIF2, REDIF3, RESTO, SCATTR, SELECT, SETADR, SETUP1, SETUP2, SETUP3, SETUP4, SETUP5, SETUP6, SETUP8, SETUP9, SET10, SET11, SET13, SET14, SET17, SET18, SET19, STODAT, STOFIL, STOMAP, TESTSP, TNSFER, TRAMTX, TRHIST, TRSTAT, WRTAMT, and WRTFIL
GRCBLK	GRAYMAP and HIST	HISTGM and HISTIC (utility subprograms); GRAYMP, HEADNG, PICT, SETUP5, and SETUP6

TABLE 5-1.- Continued.

Common block	Interfaces	
	Processor	Subprogram
GTBK	GTDDM and GTTCN	DDM, GTCRPL, GTDOTS, GTDWR, SET17, SET19, and TCN
HISTOR	GRAYMAP and HIST	HISTGM (utility subprogram); GRAYMP and SETUP6
IDSTOR	CLASSIFY, DAMRG, DATA-TR, DISPLAY, DOTDATA, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, SCTRPL, STAT, and TESTSP	MSCAN, TAPHDR, and WRTHED (utility subprograms)
IDWORD	NDHIST	RESTO
INFORM	CLASSIFY, DATA-TR, DISPLAY, DOTDATA, GTDDM, ISOCLS, LABEL, NDHIST, SCTRPL, SELECT, TESTSP, and TRSTAT	CLSCHK, CRDSTA, GRPSCN, LISTLC, PRTCOV, RDMODK, REDDAT, REDSAV, and SAVFIL (utility subprograms); ALLKIN, AVEDIV, BHTCHR, BSTCHK, CATGRY, CLRCOD, CLSFY, CLSFY1, CLSFY2, CLSMAP, CNDMAP, DATATR, DAVIDN, DOTDST, DOTS, DSPTAP, EVALSP, EVLFET, EXSRCH, FILERD, FINT1, FLDCLS, FLDFLD, FLDLAC, FLDMEN, FLDSUB, FLDTYP, GENRPT, GTSTAT, KBTRAN, KNEAR, LABLR, LISTSM, LNTRAN, MAPHDG, MAPHND, NDHST1, PRELIM, PRTFLD, REDIF2, REODER, SCATTR, SCTRPL, SELECT, SETADR, SETUP2, SETUP4, SETUP8, SETUP9, SET10, SET11, SET13, SET14, SET17, SET19, TNSFER, TRAMTX, TRHIST, TRNDIV, TRNSFR, TRSTAT, USERIN, WHRPLC, and WRTFIL
ISOLNK	CLASSIFY, DAMRG, DATA-TR, DOTDATA, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, STAT, and TESTSP	SETUP7 and TAPHDR (utility subprograms); DAMRG, DOTS, ISOCLS, ISODAT, ISOPAT, PSPLIT, PSPPAT, SET18, and TESTSP
LABS	LABEL	ALLKIN, CLSMAP, CNDMAP, DOTDST, DSPTAP, FILERD, KNEAR, LABDOT, LABLR, MANORD, MIXMAP, and SET14
LISTMM	DISPLAY	LISTSM
MRGDAT	DAMRG	DAMRG and SET18
NDIM	NDHIST	ADDRES, FLDCLS, FLDFLD, FLDMEN, FLDSUB, NDHST1, NDHST2, PICOLR, SET10, and WRTFIL

TABLE 5-1.- Concluded.

Common block	Interfaces	
	Processor	Subprogram
PASS	ISOCLS and TESTSP	CHAIN, CLDIST, DSTAPE, PRINT, RDMEAN, and SETUP7 (utility subprograms); COVAR1, COVPAT, ISOCLS, ISODAT, ISOPAT, PSPLIT, PSPPAT, RDDATA, RDDPAT, and TESTSP
PASSA	ISOCLS and TESTSP	RDMEAN (utility subprogram)
PASSB	CLASSIFY, DATA-TR, ISOCLS, LABEL, SCTRPL, SELECT, TESTSP, and TRSTAT	CRDSTA and RDMODK (utility subprograms)
SCRACH	CLASSIFY	CLSFY, CLSFY1, CLSFY2, and RELERR
SCTTER	SCTRPL	CLRCOD, CNTER, LINPLT, OFFSET, RESCLE, SCATTR, SCTRPL, SETADR, SET11, STOFIL, TNSFER, and UNPCKV
STBASE	STAT	LEARN, SETUP1, and STAT
STCBLK	STAT	LEARN, SETUP1, and STAT
TAPERD	CLASSIFY, DAMRG, DATA-TR, DOTDATA, DISPLAY, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, STAT, and TESTSP	FLDINT, LINERD, MSCAN, SEARCH, TAPHDR, and WRTHED (utility subprograms); DAMRG, DDM, SET17, SET18, SET19, and TCN
TR	GTDDM	GTDTL, GTDWR, GTTRNS, and SET19
TRBLCK	DATA-TR and TRSTAT	PRTCOV (utility subprogram); DATATR, KBTRAN, LNTRAN, MAXMAT, SETUP8, TRANSF, and TRHIST
WRTAP	DAMRG, DATA-TR, DISPLAY, GTTCN, ISOCLS, LABEL, SCTRPL, and TESTSP	WRTHED and WRTLN (utility subprograms); DAMRG, SET18, and TCN

5.1 ARRAY

Common ARRAY is a block of working storage in the blank common block. It is passed to each processor and is used by every processor for the variable dimensioning of other arrays. In the Initial Program Load (IPL) version of the EOD-LARSYS, ARRAY is placed in new labeled common /BLANK/. The dimension of ARRAY currently is TOP = 10 600 four-byte words.

5.2 BESTKN

Common BESTKN is used only in the SELECT processor. Its parameters are defined as follows:

KPPPTS(60) — KPPPTS(I) = number of points in the *ith* subclass after grouping.

IPRIOR — Switch to activate the a priori weighting option.

KBEST — Value for k in the option including the best k of n passes.

NCPASS — Number of channels per pass, usually four.

5.3 BMTRX

Common BMTRX is used by the CLASSIFY processor to store the B-matrix. The parameter BMATRX(450) is set in the SETUP2 subprogram.

5.4 BUFF

Common BUFF is used by all processors that call the LINERD subprogram; that is, CLASSIFY, DAMRG, DATA-TR, DOTDATA, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, STAT, and TESTSP. The parameter IBUFF(3375) is used to store the results of the current line read from subprogram LINERD.

5.5 CLASS

Common CLASS is used by the CLASSIFY processor and is in core only when this processor is executing. Its parameters are defined as follows:

APRFLG — Number of a priori values read from input card images.

BMCOMB — Number of linear combinations in the B-matrix.

BMFEAT — Number of channels used in computing the B-matrix.

BMFLG — Flag indicating whether the B-matrix has been input.

NOCAT — Number of categories to process.

THIJ1 — Beginning address for storing the class-pair threshold table.

IDATA1 — Beginning address for storing data.

NFILE — Number of the next file to be written on the MAPTAP file.

STATKY — Flag controlling the statistics printout.

CATNAM(60) — Contains category names.

CLSSYM — Default symbols used in printing the classification map.

CON(60) — Contains subclass constants.

DET(60) — Contains subclass determinants.

FLDESC — Field name.

FLDINF(6) — Contains rectangular coordinates surrounding the nonrectangular field.

KCLSNA(60) — Contains the class names in the order taken off the CATEGORY control card image.

NOCTCL(60) — Number of classes in each category.

SUBCAT(60) — Contains the category number to which the corresponding subclass belongs.

NOCHAN — Number of channels to be processed.

CHNVEC(30) — Vector of channel numbers.

5.6 <u>DISPL</u>

Common DISPL is used only by the DISPLAY processor. The parameter definitions are as follows:

CATFLG — Flag indicating whether or not category performance reports must be generated.

CATNAM(61) — Names of categories read from the MAPTAP file.

CLSNAM(61) — Names of classes read from the MAPTAP file.

SUBNAM(61) — Names of subclasses read from the MAPTAP file.

SUBNO(60) — Subclass numbers read from the MAPTAP file.

SUBCAT(60) — Subclass/category correspondence vector; SUBCAT(I) = M means subclass i belongs to category m.

CLSSUB(60) — Subclass/class correspondence vector; CLSSUB(I) = M means subclass i belongs to class m.

NOMAP — Key indicating whether or not a classification map is to be printed.

TOTVT3 — Total vertices in input test fields.

NOSUB3 — Number of subclasses used in CLASSIFY plus one for the threshold class.

PCFDKY — Key indicating whether or not ground-truth performance reports are to be printed on a per-field basis.

TSTKEY — Key indicating whether or not test fields were input.

TRNKEY — Key indicating whether or not training fields are to be outlined.

THRSKY — Threshold key: = 1, chi-squared thresholds are applied; = 2, empirical thresholds are applied; = 3, user-input thresholds are applied; = 4, Fisher F-distribution thresholds are applied; and = 0, no thresholding.

STATKY — Key for printing statistics from the MAPTAP file.

EMPTRS — Empirical thresholding flag.

THRSVA — User-input threshold flag.

PLTKEY — Flag for printing cumulative histograms of the quadratic form.

BMFLG — Flag indicating whether or not the B-matrix was applied in the CLASSIFY processor.

BMCOMB — Number of linear combinations in the B-matrix.

BMFEAT — Number of channels used in computing the B-matrix.

CDATE(2) — Date of classification.

FLDSV2 — Address in ARRAY for training field information. For each training field, four items of information are stored: 1 = field name, 2 = class number, 3 = subclass number, and 4 = number of vertices.

FIELD2 — Address in ARRAY for rectangular area surrounding each training field. For each training field, five items of information are stored: 1 = line start, 2 = line end, 3 = sample start, 4 = sample end, and 5 = pointer into VERTX2 array for vertices of the field being processed.

VERTX2 — Address in ARRAY for training field vertices.

FLDSV3 — Address in ARRAY for test field information. For each test field, four items of information are stored: 1 = field name, 2 = class number, 3 = subclass number, and 4 = number of vertices.

FIELD3 — Address in ARRAY for rectangular area surrounding each test field. For each test field, five items of information are stored: 1 = line start, 2 = line end, 3 = sample start, 4 = sample end, and 5 = pointer into VERTX3 array for vertices of the field being processed.

VERTX3 — Address in ARRAY for test field vertices.

PCTID3 — Address in ARRAY for classification performance table.

THRES(60) — Array containing threshold values.

SYMMTX(66) — Symbols for each subclass, including threshold and outline symbols.

HIGH(60) — Threshold rejection percentage for the empirical option.

CON(60) — Constant factor from the probability density function (PDF) in CLASSIFY (one for each subclass).

FLDKEY — Key indicating whether ground-truth fields are associated with classes or subclasses.

NOFLD2 — Number of training fields.

NOFLD3 — Number of test fields.

NOFET2 — Number of channels used in classification.

FETVC2(30) — Channels used in classification.

NOSUB2 — Number of subclasses used in classification.

NOTRFD — Number of ground-truth fields for which performance tables will be made; = NOFLD2 or NOFLD3.

TOTVT2 — Total number of vertices for training fields.

NOCLS2 — Number of classes used in classification.

KATNO(60) — Class/category correspondence vector; KATNO(I) = M means class i is in category m.

NOCAT — Number of categories used in classification.

FILTER — Flag for spatial filtering option.

MAPFMT — Format for output MAPUNT file.

DESKEY — Key indicating whether or not designated fields were input.

DESUNI — Number for DU (NOSUB2 + 5).

DESOTH — Number for DO (NOSUB2 + 6).

CROP — Name of crop for which ITS summary report is to be printed; crop is to be compared with "other."

ACROP — Acres of crop (user input).

AOTHER — Acres of "other" (user input).

ATOTAL — Total acres in classified segment.

SITE(6) — Name of site (classified segment).

ANALYS(5) — Name of analyst performing study.

CAM(15) — Name of procedure configuration used in study.

CRPKEY — Key for generating ITS summary report.

KEPPTS(60) — Total number of pixels in each subclass.

DOTKEY — Key indicating whether or not dot data classification performance summaries are to be processed; = 0, no dot data processing; > 0 (and DOTERR = 0), dot performance summaries are provided.

DOTERR — Flag indicating that the dot data processing will be discontinued either because of disagreement between labeled and classification category names or because of erroneous parameters on the DOTFILE control card image.

5.7 DOTVEC

Common DOTVEC is used in the extraction of dot information from user-input card images to the DISPLAY and DOTDATA processors. Its parameters are defined as follows:

TYPE — Type (1 or 2) of last dot read.

CATNAM(60) — Category names indexed by NOCAT for type 1 or 2 dots.

NOCAT — Number of unique categories for type 1 or 2 dots.

TOTVEC — Total number of type 1 or 2 dots.

FLDINF(6) — Line and sample numbers of the field for the dot currently being read.

PRTKEY — Switch to activate the printer.

SIZE — Size of dot record (4 + NOFET2) to be written on the DOTUNT file.

LACIE — Option to read dot data card images in Earth Resources Interactive Processing System/Large Area Crop Inventory Experiment (ERIPS/LACIE) format; switch to indicate input card images will be in ERIPS/LACIE format.

5.8 DVNBLK

Common DVNBLK is the storage area for numbers computed during execution of the Davidon-Fletcher-Powell option in the SELECT processor. It is used only by the SELECT processor, and results are stored in the following parameters:

DFDK — Value of the function to be minimized, minus the norm of the gradient with respect to the square of the H-matrix.

CAYMIN — Minimum value of the function (XMIN2 = CAYMIN).

FII — Initial value of the function to be minimized.

CCAY — Minimum value for k (CAYMIN = CCAY).

I1D I1DMEN	Numbers computed during execution of the Davidon-Fletcher-Powell procedure.*

ITT — Counter denoting the number of evaluations of the function to be minimized.

ICNT — Number of iterations of the Davidon-Fletcher-Powell procedure.

N — Number of variables.

---

*Not defined specifically because of the complicated nature of these calculations.

5.9 FNTDUM

Common FNTDUM is used only by the SELECT processor. The parameters are defined as follows:

ITT — Initialized in call to subprogram FINT1 and reserved for future use.

ICYCLE — Counter for Davidon-Fletcher-Powell iterative technique implementation.

5.10 FSL

Common FSL is used only by the SELECT processor. The parameters utilized in FSL are as follows:

CFAC — Constant factor needed to compute average divergence.*

TOTMSR — Separability measure computed using all channels.*

SEPMSR — Separability measure computed using best set of channels.

PRCKEY — Key indicating which procedure to execute: = 1, Exhaustive Search; = 2, Without Replacement; = 3, Davidon-Fletcher-Powell; = 4, user-input B-matrix evaluation; and = 5, user-input channel set.

CRIKEY — Key indicating which criterion is to be used: = 1, weighted average divergence; = 2, weighted average transformed divergence; and = 3, weighted average Bhattacharyya distance.

INCFET — Number of channels to include in the best set (user input via the INCLUDE control card).

INCVEC(30) — Vector containing the channels to be included (user input via the INCLUDE control card); meaningful only if the Without Replacement procedure is executed.

ICOUNT — Maximum number of iterations of the Davidon-Fletcher-Powell procedure.

SETWGT — Key indicating whether or not default weights are to be set.

EVALBF(100) — Buffer containing all user "evaluate" requests;
EVALBF(1) = number of channels for first request;
EVALBF(2 - N) = a set of channels to be evaluated;
EVALBF(N + 1) = number of channels on second request, etc.

---

*Double precision in subprogram PRELIM.

FETVC4(30) — Vector containing the set of channels selected as the best set.

NOFET4 — Dimension of FETVC4.

VARSZ4 — Size of the covariance matrix for the best set $\left[\text{NOFET4}\left(\frac{\text{NOFET4} + 1}{2}\right)\right]$.

CORBAS — Next available address in ARRAY for storage of data.

DTAB4 — Beginning address in ARRAY for interclass separability measure table.

WGHS14 — Beginning address in ARRAY for interclass weights.

BESTVC(10) — Vector containing the best request from user input.

DIVSIZ — Size of interclass separability measure table.

STATKY — Key indicating whether or not to print statistics.

The following variables are beginning addresses for five scratch input/output areas on the random-access drum file.

ADRESD — Drum address for storage of interclass separability measures computed using all channels.

$$\left.\begin{array}{l}\text{ADRESP}\\ \text{ADRESF}\\ \text{ADRSH1}\\ \text{ADRSH2}\end{array}\right\}$$ Addresses for temporary storage of data arrays in the Davidon-Fletcher-Powell procedure only.

5.11 GLOBAL

Common GLOBAL is used in every processor and is always in core. All parameters are initialized in a block data subroutine called by the MONTOR or MONPAC executive routine, except as noted in the following definitions:

HEAD(63) — Standard heading printed on most output pages.

MAPTAP — Number of the Fortran unit on which the classification map is written (=2).

DATAPE — Unit number for the MSS image data tape (=3).

SAVTAP — Number of unit on which the statistics file is written (=1).

BMFILE — Number of unit on which the B-matrix file is written (=10).

BMKEY — Key indicating that the B-matrix file has been written; can be set in the SELECT, CLASSIFY, or DATA-TR processor.

HISFIL — Number of unit on which the histogram file is written (=13).

HISKEY — Key indicating the histogram file has been written; set in the HIST processor.

TRFORM — Number of the unit on which the transformed image is written by the DATA-TR processor (=14).

ERIPTP — Number of unit on which the ISOCLS or DISPLAY processor has written cluster statistics for the ERIPS (=15).

ERPKEY — Key indicating that the ERIPS interface tape has been written.

MAPUNT — Number of the unit on which the ISOCLS or DISPLAY processor writes the clustered or classified data to be displayed (=16).

NOFILE — Number of files written on the MAPUNT; set in either the ISOCLS or DISPLAY processor, which outputs the MAPUNT.

DRUMAD — Beginning address for the random-access high-speed disk file; used as a scratch file in several processors accessed by references to subprograms RREAD and RWRITE, set = 1.

DRMWDS — Number of words available on the random-access disk file.

PAGSIZ — Number of lines available for printing on a page.

DATFIL — Number of end of files (EOF's) to be read by subprogram TAPHDR in order to position the data tape to the desired file.

STAFIL — Number of EOF's to skip to position the SAVTAP file.

ASAV — Number of the unit on which the TRSTAT processor writes the transformed statistics.

ASAVFL — Number of EOF's to skip to position the ASAV file.

NHSTUN — Number of the unit on which the NDHIST processor writes the N-dimensional histogrammed data (=4).

NHSTFI — Number of EOF's to skip to position the NHSTUN file.

SCTRUN — Number of the unit on which the SCTRPL processor writes scatter plot data (=12).

MAPFIL — Number of EOF's to skip to position the MAPUNT file.

DOTUNT — Number of the unit on which the DOTDATA or LABEL processor writes the DOTFIL.

DOTFIL — Number of EOF's to skip to position the DOTUNT file.

NCHPAS — Number of channels per pass.

TRNSFL — Number of EOF's to skip to position the TRFORM file.

BMTRFL — Number of EOF's to skip to position the BMFILE.

HISTFL — Number of EOF's to skip to position the HISFIL.

PCHUNT — Number of unit corresponding to the card punch (=7).

CRDUNT — Number of unit corresponding to the card reader (=21).

PRTUNT — Number of unit corresponding to the line printer (=6).

RANDIO — Number of unit corresponding to the direct-access temporary file (=22).

5.12 GRCBLK

Common GRCBLK contains the information needed by the GRAYMAP and HIST processors. Its parameters are defined as follows:

MAXFET — Maximum number of channels.

NOFEAT — Number of channels to histogram.

NOFET2 — Number of channels to graymap.

FETVEC(30) — Array containing channels to histogram.

FETVC2(30) — Array containing channels to graymap.

FLDINF(6) — Array containing field information.

INFMT — Format of input data.

FILESV — Key to indicate if more than 50 fields have been input to the HIST processor.

NOHIST — Number of channels to display in the HIST processor.

HISVEC(30) — Array of channels to display in the HIST processor.

NOFLD — Number of fields.

FLDPTS — Number of points in the field.

XSIZ — XHGH minus XLOW.

XLOW — Minimum x-value to be histogrammed.

XHGH — Maximum x-value to be histogrammed.

YSIZ — Height of y-axis in the histogram.

5.13 GTBK

Common GTBK is used by both the GTDDM and GTTCN processors. The parameter definitions are as follows:

NRDR — Number of unit on which control card images are input; set = CRDUNT.

NPRT — Number of unit on which to write labeled dot data; set = PRTUNT.

PRTKEY — Key that determines the mask; = 1, transition year; = 2, phase 3; or = 3, user input.

VLB(6) — In GTTCN, used to hold six values corresponding to subpixels for use in labeling pixels; in GTDDM, used to hold input transformation from control card relating crop code elements to a range of numbers.

GTRDU — In GTDDM, input ground-truth unit as prepared by GTTCN; in GTTCN, input unit of ground-truth data.

GTRDF — In GTDDM, file number of input (default = 1, six subpixels per pixel); in GTTCN, input file number (default = 1, six subpixels per pixel).

GTWRU — Output unit number of GTTCN for ground-truth classification map.

GTWRF — Output file number of GTTCN.

GTNOF — Number of ground-truth files to process.

5.14 HISTOR

Common HISTOR is used as a switch in the GRAYMAP and HIST processors. The parameter HF = 1 for execution of GRAYMAP and = 0 otherwise.

5.15 IDSTOR

Common IDSTOR is used for storage of LARSYS-III-formatted* data input via subprogram TAPHDR. It is used as an interface between subprograms MSCAN, TAPHDR, and WRTHED. Its parameter IDD(250) is used to store data from the header record which are in LARSYS III format.

*The terms LARSYS II and LARSYS III are used interchangeably in this document, inasmuch as very little difference exists between the two and both formats are compatible with the EOD-LARSYS.

5.16 IDWORD

Common IDWORD is used only by the NDHIST processor. The parameter IDWORD(1000) holds one line of classification results, which is read from the direct access file upon call to subprogram RESTO. Subsequent calls to entry point RESTOR extract data from IDWORD for use by subprogram NDHST1. (This common block is needed because, in IBM Fortran, local variables are not preserved after return from a subroutine.)

5.17 INFORM

Common INFORM is used by the CLASSIFY, DATA-TR, DISPLAY, DOTDATA, GTDDM, ISOCLS, LABEL NDHIST, SCTRPL, SELECT, TESTSP and TRSTAT processors. The subprogram REDSAV, which is called during the execution of these processors, reads the SAVTAP file and reduces the statistics to a subset requested by the user via input of a CHANNELS, SUBCLASS, or GROUP control card. The reduced statistics are stored in ARRAY, and the base addresses are returned in common INFORM. The base addresses are as follows, with the size of each item stored in ARRAY given in parentheses.

NOCLS2 — Number of classes from the SAVTAP file to be used for processing.

NOSUB2 — Number of subclasses from the SAVTAP file to be used for processing.

NOFET2 — Number of channels to be used for processing.

VARSZ2 — Size of each covariance matrix $\left[\text{NOFET2}\left(\frac{\text{NOFET2} + 1}{2}\right)\right]$.

TOTVT2 — Number of vertices for training fields.

NOFLD2 — Number of training fields.

AVAR2 — Base address for means (NOFET2 × NOSUB2).

COVAR2 — Base address for covariance matrices (VARSZ2 × NOSUB2).

CLSID2 — Base address for class names (NOCLS2).

SUBNO2 — Base address for the number of subclasses in each class.

SUBDS2 — Base address for subclass names (NOSUB2).

FLDSV2 — Training field information (4 × NOFLD2); for each field, 1 = field name, 2 = class number, 3 = subclass number (0 if none), and 4 = number of vertices.

VERTX2 — Base address for vertices (2 × TOTVT2).

FETVC2(30) — Vector of channel numbers to be used for processing.

SUBVC2(75) — Vector of subclass numbers to be selected for processing.

SUBPTR(75) — Vector of corresponding subclasses (those read from the SAVTAP file and those used in CLASSIFY). SUBPTR(15) = 10 means the 15th subclass on the SAVTAP file will be the 10th subclass used in classification.

CLSVC2(60) — Vector of class numbers to which corresponding subclasses belong.

KEPPTS(60) — Number of pixels in each subclass.

NOGRP — Number of group card images input by user.

The following parameters are set up by the GRPSCN subprogram, which is called by the setup routines in the CLASSIFY and SELECT processors when a GROUP control card image is read. This grouping is the combining of two or more subclass statistics to form one set of subclass statistics.

GRPNAM(60) — Names for the combined statistics.

GRPDEX(61) — Index array pointing to the start of the corresponding group in the GROUPS array.

GRPCHK(61) — Vector designating if a subclass has been assigned to a group.

GROUPS(124) — Array containing subclasses within a group.

5.18 ISOLNK

Common ISOLNK is used by the CLASSIFY, DAMRG, DATA-TR, DOTDATA, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, STAT, and TESTSP processors, primarily to extract information from the header record of Universal-formatted files. The parameters are defined as follows:

SUNANG(8) — Vector containing Sun angles.

ISUNT — Switch (≠0) to extract Sun angles from the header record.

ISUNC — Switch to obtain Sun angles from user-input card images.

SMSTR — Starting sample number from the header record.

SMSTP — Ending sample number from the header record.

SMINC — Sample-skip factor from the header record.

LINSKP — Line-skip factor from the header record.

5.19 LABS

Common LABS is used only by the LABEL processor. The parameters are defined as follows:

NOCAT — Number of unique categories from the DOTUNT file for type 1 or 2 dots.

CATNAM(60) — Category names from the DOTUNT file for type 1 or 2 dots.

NOCL2 — Number of classes.

CLSNM2(60) — Contains class names.

NOCAT2 — Number of categories.

CATNM2(60) — Contains category names.

SUBRAY(120) — Array of subclass numbers.

PRT(60) — Pointers used in relabeling clusters.

CATPTR(250) — Pointers for dot relabeling.

CATDOT(500) — Dot category numbers (types 1 and 2).

DOTVEC(250) — Array of dot information for the field being processed.

COND — Key to print conditional cluster map.

MIX — Key to print mixed cluster map.

PROC — Key for selection of labeling option: = 1, k-nearest-neighbor; = 2, all-of-a-kind; and = 3, SAVTAP or DOTFIL relabeling option.

MAPKEY — Key (≠0) to read the MAPUNT file and store the image on high-speed disk.

DOTKEY — Key (≠0) to read a file from the DOTUNT and print saved training or test fields.

STATKY — Key (≠0) to read a file from the SAVTAP and print statistics.

SUNANG — Sun angle.

T — Threshold value in determining conditional clusters.

NEARST — Number of k-nearest dots.

DIST — Distance norm in computing cluster-dot distances: = 1, use L1; and, = 2, use L2 (Euclidean).

NOFEAT — Number of features on DOTFIL.

FETVEC(30) — Array of channels to [illegible] used in processing.

OMAPUN — Output unit number for displaying interface file.

OMAPFI — Output relative file number of OMAPUN.

OSAVTP — Unit number for output SAVTAP file.

OSTAFI — Output file number of OSAVTP.

NOSUN — Number of Sun angles on the DOTFIL.

ANGLE(8) — Sun-angle values on the DOTFIL.

SIZE — Size of dot record (4 + NOFET2).

TOTDT2 — Number of dots.

FLDINF(6) — Line and sample numbers of the field for the dot being processed.

CLSSYM(62) — Array of symbols for each class to be used in printing the MAPTAP file.

STADRS — DRUMAD + TOTPIX.

MEANAD — Beginning address for means.

TABADR — Beginning disk address for distance table.

MAPADR — Beginning disk address for (NSAMP - 110) points of the conditional or mixed cluster map.

SUNCOR(30) — Array of Sun-angle correction factors.

ODOTUN — Unit number of relabeled DOTFIL.

ODOTFI — File number of ODOTUN.

MANSTA — Switch to relabel the SAVTAP file.

MANDOT — Switch to relabel the DOTFIL.

DSPUNT — Unit number of conditional or mixed cluster map.

DSPFIL — File number of DSPUNT.

DSPKEY — Switch to output display interface file.

PRNSTS — Switch to print statistics.

PRNDOT — Switch to print dot data.

FLDNAM — Field name.

VERTEX(22) — Array of vertices.

NOVRT — Number of vertices.

NSUN — Number of Sun-angle correction factors.

ANGLES(8) — Vector of Sun angles.

TOTDT3 — Number of dots to be excluded.

FLDADR — Beginning disk address for field information.

VTXADR — Beginning disk address for field vertices
[TOTAL + 4(NOFLD)].

5.20 LISTMM

Common LISTMM is used only by the DISPLAY processor. The parameter definitions are as follows:

NPGA(3,2) — Number of dots in each dot data file [P = Patterson-Pitt-Thadani Classification (PPTC) file (PPUNIT); G = ground-truth file (GTUNIT); and A = analyst-interpreter (AI) file (AIUNIT)] for each dot type (dot type is 1 or 2).

NAMPGA(209,3,2) — Contains class names for each dot of each file for each dot type.

LINPGA(209,3,2) — Contains line number for each dot of each file for each dot type.

SAMPGA(209,3,2) — Contains sample number for each dot of each file for each dot type.

DOTLAB(209,4,2) — Contains label for each dot of each file for each dot type; 4 is the classification label index in DOTLAB.

VPGA(3) — Switch indicating which of the three files (PPUNIT, GTUNIT, or AIUNIT) have been input.

IPGA — Number of input file types; if = 2, then two of the three files have been input.

5.21 MRGDAT

Common MRGDAT is used only by the DAMRG processor. The parameter definitions are as follows:

IMOPT — Option switch: = 1, channel merge; = 2, spatial merge; and, = 3, pseudomerge.

ISOPT — Sun-angle correction switch; = 0, no Sun-angle correction is applied to output pixels.

NUMFIL — Number of input files; determined by counting input DATAPE control cards.

IDATTP(6) — Numbers of input units.

IDATFL(6) — Numbers of input files.

NOFEAT — Number of channels for output ,file.

NFEAT(6) — Number of channels to be considered for input files.

FETVEC(30,6) — Channel numbers for input files.

ISUN(8,6) — Sun angles of interest from input files.

SUNCOR(30) — Sun-angle correction factor (floating point) for output pixels.

FLDINF(6,6) — Rectangular field description for input files (starting line number, ending line number, line-skip factor, starting sample number, ending sample number, sample-skip factor).

NOSAMP — Number of pixels per scan line on output file.

NOLINE — Number of lines on output file.

NSS(6) — Number of samples per line on input fields in spatial option.

NACROS — Number of fields in horizontal direction in spatial merge option.

NLINES(6) — Number of lines from input files in pseudomerge option.

LINPTR(7) — Pointer to LINES.

LINES(600) — Line numbers from input files in pseudomerge option.

FORMM — Format of output file: = 1, Universal; = 2, LARSYS III.

5.22 NDIM

Common NDIM is used only by the NDHIST processor. The parameters are defined as follows:

NCLRCH — Number of color channels.

CLRVEC(30) — Vector of color channels.

MAXVEC — Maximum number of vectors that can be stored.

MAPKEY — Key indicating that a MAPUNT file has been written and will be input.

CLASS — Key indicating that fields will be grouped on a class basis.

SUBCLS — Key indicating that fields will be grouped on a subclass basis.

FIELD — Key indicating that fields will be grouped on a field basis.

MEANSW — Key indicating that means for input fields will be computed.

NOVEC — Number of the unique vector histogrammed.

FLDINF(6) — Contains rectangular field coordinates of input fields.

SIZE — Number of computer words required to store a packed histogrammed vector (NOFET2/4).

TOTMNS — Total number of elements in MEANS array.

CNTR1 — Address for storing frequency codes in ARRAY.

CNTR2 — Beginning disk address for storing frequency codes.

ID1 — Address for storing identification (ID) codes in ARRAY.

ID2 — Beginning disk address for storing ID codes.

COLOR1 — Address for storing color codes in ARRAY.

COLOR2 — Beginning disk address for storing color codes.

BUFLEN —	Amount of storage available for ID and/or color codes.
ID3 —	Cumulative ID code disk address.
COLOR3 —	Cumulative color code disk address.
NODUMP —	Number of times that ID and/or color codes were dumped onto disk.
IDATA1 —	Address for storing imagery data in ARRAY.
TOTVEC —	Total number of vectors in the area histogrammed.

5.23 PASS

Common PASS is used by the ISOCLS and TESTSP processors. The parameters are defined as follows:

STOP	Maximum number of iterations for the clustering procedure.
LNCAT	Current number of clusters.
NMIN	Minimum number of points to allow per cluster.
KRN	Number of iterations between cluster summaries (a cluster summary is printed every KRN iterations).
STDMAX	Standard deviation for splitting clusters.
DLMIN	Minimum distance between clusters for combining.
SEP	Distance to separate clusters.
MAP	A cluster map will be printed every MAP iterations.
SPTRIG	Indicates whether or not SEP was input.
IRD	Number of records to read from data file.
KPTS	Number of points in last record read.
NOPTS	Number of points in each record.
PUNCH	Indicates whether or not to punch the module STAT file.
ICHN	Indicates whether or not chaining is to be performed.
CHNTHS	Minimum distance between clusters for chaining.
ICHAIN(62)	Contains chained cluster numbers.
NWDS	Number of words available for disk storage of image data to be clustered.
IBEGIN	Beginning disk file address for input initial cluster centers.
BEGIN1	Beginning disk file address for image data.
BEGIN2	Beginning drum file address for IPLACE (cluster to which corresponding point belongs).

BEGIN3 — Beginning disk file address for temporary storage of class statistics.

CLSNAM — Name of class currently being processed.

NOFLD — Number of fields input for this class.

IPT — Number of words of storage used in ARRAY for field and class information for this class.

TOTWRD — Total words written on disk file beginning at address BEGIN1.

TOTPTS — Total points to be clustered for the current class.

NCLASS — Number of classes to be clustered for current call to ISOCLS or TESTSP.

NOCLS — Current class number.

TOTSUB — Total subclasses (clusters) for this call to ISOCLS or TESTSP.

TOTFLD — Total fields for all classes.

TOTVRT — Total vertices for all fields.

NOCL — Number of classes since last call to SETUP7.

NVRT — Number of vertices for all fields in current class.

NXTCLS — Name of next class.

NOFEAT — Number of channels used in clustering.

MAXCLS — Maximum number of clusters per class.

FETVEC(30) — Vector of channels used in clustering.

SYMMTX(62) — Symbols for printing classification map.

VARSIZ — Size of the covariance matrix for each cluster.

STATKY — Flag for printing covariances.

ISOKEY — Flag indicating that cluster centers will be read from a card image file.

MAPFMT — Format for output MAPUNT file.

MAPKEY — Indicates whether to use mean vector or cluster number in creating a MAPUNT file.

SEQUEN(20) — Array containing sequence of "S" and "C" characters for split/combine iteration control.

PERCEN — Cutoff percentage for stabilized clusters in initial sequence.

SIMERP — Key for simulating ERIPS clustering algorithm; if SIMERP = 1, simulate ERIPS; if SIMERP = 0, standard ISOCLS or TESTSP.

IORDER — Indicates color keys will be ordered based on the ranking of greenness of each cluster, as computed by subroutine RANK.

INUNIT — Unit number for input SAVTAP file.

INFILE — File number of input SAVTAP file.

INITM — Trigger to extract initial means from SAVTAP file.

PMIN — Population minimum for each cluster.

SUBVEC(62) — Vector of subclass numbers.

NOSUB2 — Number of subclasses from the SAVTAP file to be used in processing.

CHNVC(30) — Vector of channels to be used in processing.

NOCHAN — Number of channels to be used in processing.

ERCOMP — Number of iterations between special printout showing progress of clustering.

NOSEQ — Sequence number of iteration.

MEANDO — Mean of a DO cluster.

MEANDU — Mean of a DU cluster.

SYMDO — Classification map symbol for a DO field.

SYMDU — Classification map symbol for a DU field.

ITRIGO — Indicator that a DO field has been read.

ITRIGU — Indicator that a DU field has been read.

DOFLAG — Indicator (=1) that DO pixels are in the class fields stored in RDDATA or RDDPAT.

DUFLAG — Indicator (=1) that DU pixels are in the class fields stored in RDDATA or RDDPAT.

DODU — DOFLAG + DUFLAG.

STDOTS(60) — Vector of starting dot numbers.

NSDOTS — Number of starting dots used in LACIE Procedure 1 processing.

SUNCOR(30) — Sun-angle correction factors.

LLNCAT — LNCAT + DODU.

DVERT(250,2) — Vertices of DO/DU fields.

DRECT(60,2) — Information describing enclosed rectangles of DO/DU fields.

DVPNT(11,2) — Pointer to DO/DU field vertices.

IDCNT(2) — Number of rectangles surrounding DO/DU fields.

NDOU(2) — Numbers of DO/DU pixels in the class fields.

MXFET1 — Maximum possible number of channels (=30).

MAXPOP — Maximum possible number of regular clusters + DO/DU clusters (=62).

5.24 <u>PASSA</u>

Common PASSA has the following parameters for use by the ISOCLS and TESTSP processors:

NOFET1 — Number of features input on card images from the mean deck.

FTVEC1(30) — Array of feature numbers input on card images.

5.25 PASSB

The CLASSIFY, DATA-TR, ISOCLS, LABEL, SCTRPL, SELECT, TESTSP, and TRSTAT processors use common PASSB to store some of the information extracted from the module STAT file. The parameters are defined as follows:

NOCLS — Number of classes.

NOSUB — Number of subclasses.

NOFEAT — Number of features.

NOFLD — Number of fields.

TOTVRT — Total number of field vertices.

FETVEC(30) — Array of channel numbers.

FLDSV1 — Pointer to field information computed in subprogram CRDSTA.

CLSID1 — Pointer to class names computed in subprogram CRDSTA.

5.26 SCRACH

Common SCRACH is the buffer for the CLASSIFY processor. The parameter IDATA(12500) is the general scratch area during execution of this processor.

5.27 SCTTER

Common SCTTER is used only by the SCTRPL processor. The parameters containing SCTRPL data are defined as follows:

RSCALE — Key indicating that the transformed data are to be rescaled.

XYSCLE — Key indicating that the pixel frequency plot data are to be rescaled to a specified range.

CLRVEC(30) — Vector of color channels.

NCLRCH — Number of color channels.

CLRKEY — Key indicating the manner in which colors are being defined: = 1, input by SAVTAP file; = 2, user input; = 3, radiance values from image tape stored on NHSTUN file; and = 4, field means stored on NHSTUN file.

LOG — Key indicating LOG(2) of frequency is to be output on the line printer plot.

FREQ — Key indicating frequency is to be output on the line printer plot.

XMAX — Maximum value of first component of transformed data.

YMAX — Maximum value of second component of transformed data.

XMIN — Minimum value of first component of transformed data.

YMIN — Minimum value of second component of transformed data.

BCKGND — Color for tape background.

XHI — Upper limit of scan-line parameter.

XLO — Lower limit of scan-line parameter.

YLO — Lower limit of line number parameter.

XSIZ — Number of samples per scan line.

YHI — Upper limit of line number parameter.

YSIZ — Number of lines to output on tape.

NBINS — Number of bin levels or symbols for pixel frequency plot.

SYMMTX(32) — Contains symbols for pixel frequency plot.

BMATRX(60) — Contains B-matrix.

BVEC(30) — Contains additive vector.

NBVCHN — Number of additive vector elements.

NOFEAT — NOFET2 + NCLRCH.

SCALKY — Key indicating the manner of collecting the minimum and maximum values: = 1, user input; or = 2, computed from the NHSTUN file.

MENADR — Address for storing means.

FLDADR — Address for storing field information.

PNTADR — Address for storing point values.

IDADR — Address for storing ID codes.

NC — Number of channels for colors to be output on tape.

BMFEAT — Number of channels in B-matrix.

BMCOMB — Number of linear combinations in B-matrix.

NOVEC — Number of vectors on the NHSTUN file.

TOTMNS — Number of mean elements.

SIZE — Number of words for packed, histogrammed vector.

DRMID — Beginning disk address for storing ID codes.

DRMID1 — Summing disk address for storing ID codes.

DRMCLR — Beginning disk address for storing colors.

DRMCR1 — Summing disk address for storing colors.

DRMTNS — Beginning disk address for storing transformed data.

DRMTN1 — Summing disk address for storing transformed data.

DRMCNT — Beginning disk address for storing frequencies.

DRMCT1 — Summing disk address for storing frequencies.

DRMVEC — Beginning disk address for storing vectors.

DRMVC1 — Summing disk address for storing vectors.

VECTR1 — Address in ARRAY for storing transformed vector.

DATA1 — Same as VECTR1; used for creating a scan line of data output to SCTRUN.

NVEC — Number of vectors to read from disk at one time.

NOREAD — Number of reads to disk.

LREAD — Number of vectors to read on last disk read.

DRMPTR — Beginning disk address for storing pointers.

DRMPT1 — Summing disk address for storing pointers.

FETVEC(16) — Contains FETVC2 and CLRVEC channels.

DRMPLT — Beginning disk address for storing frequency plot image.

CSCALE — Key indicating that XSIZ or YSIZ parameters have been input.

NOSUB — Number of input subclasses.

5.28 STBASE

Common STBASE is used by the STAT processor. It contains the base addresses for the statistics stored in ARRAY. The parameters are defined as follows:

SUBSV1 — Base address for subclass information (5 × SUBNO): For each subclass index, 1 = class number, 2 = starting field number, 3 = ending field number, and 4 = subclass name.

SUBMN1 — Base address for subclass means.

SUBVR1 — Base address for subclass variances.

SUBSD1 — Base address for subclass names.

SUBCL1 — Base address for class number.

SAVER1 — Base address for training field vertices.

HSTAL1 — Base address for subclass histogram totals.

SPEC1 — Base address for spectrogram information (5 × NOSPEC).

COVAR1 — Base address for field covariances.

AVAR1 — Base address for field means.

CLSID1 — Base address for class names.

FLMEN1 — Base address for field means.

FLVAR1 — Base address for field variances.

HFTAL1 — Base address for field histogram totals.

FLDSV1 — Base address for field information (10 × MAXFLD): For each field index, 1 = field name, 2 = class number, 3 = subclass number, 4 = number of vertices, 5 = starting line number, 6 = ending line number, 7 = starting sample number, 8 = ending sample number, 9 = line increment, and 10 = sample increment.

5.29 STCBLK

Common STCBLK contains information needed by routines in the STAT processor, as defined in the following parameters.

MAXFET — Maximum number of channels.

MAXCLS — Maximum number of classes.

MAXFLD — Maximum number of fields.

NOFEAT — Number of channels requested.

NOFET2 — Number of channels input.

VARSIZ — Size of each covariance matrix $\left[\text{NOFEAT}\left(\frac{\text{NOFEAT} + 1}{2}\right)\right]$.

NOSPEC — Number of groups of subclasses to plot.

NOHIST — Number of channels to histogram.

SPCBAS — Mi imum radiance value on y-axis of spectral plot.

IBLOCK(30) — Contains keys to certain options in the STAT processor.

FETVEC(30) — Vector of selected channels.

FETVC2(30) — Vector of input channels.

HISVEC(30) — Vector of channels to histogram.

NOFLD — Number of fields.

NOCLS — Number of classes.

FLDINF(6) — Field information.

FLDPTS — Number of points in field.

CLSPTS — Number of points in class.

XSIZ — XHGH - XLOW (=101).

XHGH — Maximum x-value to be histogrammed (=255).

XLOW — Minimum x-value to be histogrammed (=0).

YSIZ — Height of y-axis in the histogram (=15).

5.30 TAPERD

Common TAPERD is used in all processors which call subprogram TAPHDR; i.e., CLASSIFY, DAMRG, DATA-TR, DOTDATA, GRAYMAP, GTDDM, GTTCN, HIST, ISOCLS, LABEL, NDHIST, STAT, and TESTSP. The parameters are defined as follows:

IUNIT — Unit number; set in the calling sequence.

IFRST — Scan-line number of the first data record.

FSCAN — Normally set equal to IFRST in the TAPHDR subprogram.

SAMEND — Ending sample number of user-specified field.

SAMINC — User-specified sample increment.

READY — Initialization parameter.

NSCAN — Starting scan-line number.

LINC — Line increment (number of records to skip between scan lines).

ID(200) — Storage for extracted data files from the header record.

DSL — (Ancillary length + number of samples) × number of channels on the data file (data set length in bytes).

$$\left.\begin{array}{l}\text{LBUF(30)}\\ \text{JREC(30)}\\ \text{IBYTE(30)}\end{array}\right\}\ \text{Storage areas for feature extraction data from the input tape.}$$

NBUFS — $\frac{\text{Number of records per data set}}{10}$.

FILENO — Relative file number on input unit.

LINEND — Ending line number of user-specified field.

LININC — User-specified line increment.

NSAMP — Number of samples.

NOCHAN — Number of user-specified channels.

FORMT — Universal or LARSYS II format.

5.31 TR

Common TR is used only in the GTDDM processor. The parameters are defined as follows:

TRNS1(256) — Storage for category names corresponding to crop code numbers; input from TRANS control card.

TRNS2(26) — Working storage.

TRNS3(26) — Storage for alphabetical crop codes found upon scanning TRNS1.

TY(11,19) — Storage for dot grid elements for type 1 and type 2 dots, according to input.

5.32 TRBLCK

Common TRBLCK is used by the DATA-TR and TRSTAT processors. The parameters are defined as follows:

OUTFMT — Format of the output TRFORM file.

NOFEAT — Number of features.

FLDINF(6) — Line and sample numbers of the field being processed.

FETVEC(30) — Vector of channel numbers.

5.33 WRTAP

Common WRTAP is used by the processors which call subprograms WRTHED and WRTLN; i.e., DAMRG, DATA-TR, DISPLAY, GTTCN, ISOCLS, LABEL, SCTRPL, and TESTSP. The parameters are defined as follows:

ICOUNT — Number of data sets written.

FORMT — Universal or LARSYS II format.

UNIT — Unit number of the file being processed.

VARBL(600) — Storage for formatted data.

IREMD — In the case of Universal format: number of channels

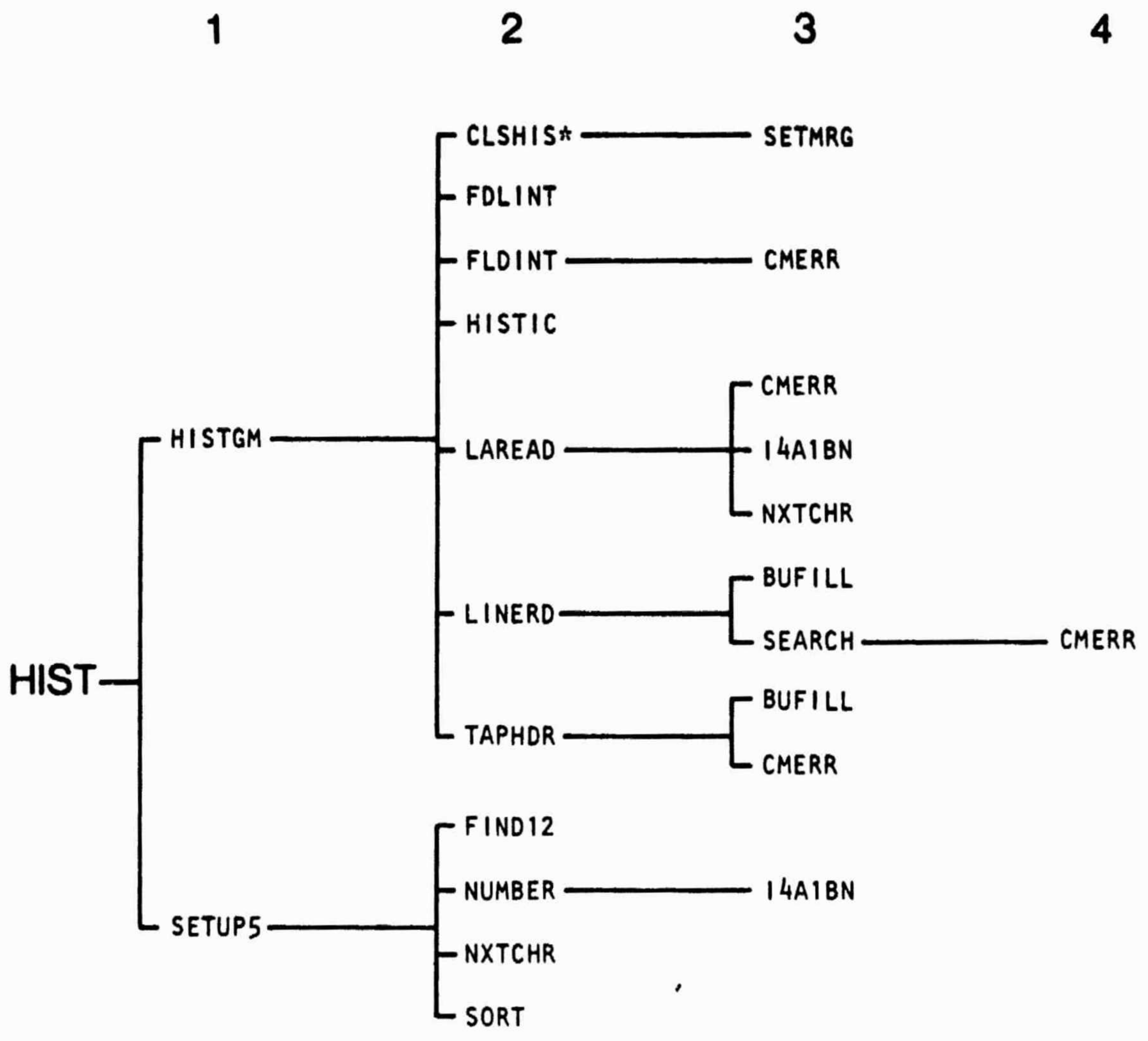
$$\times \frac{\text{number of channels + ancillary length}}{300}.$$

6. HIST PROCESSOR SUBPROGRAMS

The HIST processor computes individual field histograms and a total histogram for all fields. Only three of its subprograms are exclusive to the processor: the driver routine HIST; SETUP5, which accepts tape input and reads the control card images; and SORT, which arranges integers in ascending order. The histograms are performed and output by the utility subprogram HISTGM and the subprograms which it calls. (The utility subprograms are described in section 19.) Figure 6-1 is a linkage diagram of the HIST processor.

HIST PROCESSOR

Subroutine level



*Entry points are COMHST, FLDHIS, and HSTGRM

Figure 6-1.- Linkage diagram for the HIST processor.

6.1 HIST

The HIST subprogram is the driver routine for the HIST processor.

6.1.1 LINKAGES

The HIST routine calls the HISTGM and SETUP5 subprograms. It is called by MONTOR.

6.1.2 INTERFACES

The HIST subprogram interfaces with other routines through the calling arguments.

6.1.3 INPUTS

Calling sequence: CALL HIST(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

6.1.4 OUTPUTS

Not applicable.

6.1.5 STORAGE REQUIREMENTS

This subprogram requires 436 bytes of storage.

6.1.6 DESCRIPTION

The HIST subprogram coordinates the logical steps for histogramming data. It calls SETUP5 to read in control card images and HISTGM to perform histogramming and output the histogram file (HISFIL).

6.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 6.4.

6.1.8 LISTING

The subprogram listing is provided in volume IV, section 6.

6.2 SETUP5

The SETUP5 subprogram reads and analyzes the control card images for the HIST processor.

6.2.1 LINKAGES

The SETUP5 subprogram calls the FIND12, NUMBER, NXTCHR, and SORT subprograms. It is called by HIST.

6.2.2 INTERFACES

The SETUP5 subprogram interfaces with other routines through common blocks GLOBAL and GRCBLK and through the calling arguments.

6.2.3 INPUTS

Calling sequence: CALL SETUP5(FILHS,FLDTL,TOTTL,TOP)

Parameter	Dimension	In/out	Definition
FILHS	1	Out	Constant used in calculating FLDTL.
FLDTL	1	Out	Number of fields to be histogrammed.
TOTTL	1	Out	Size of field table.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

The control card images relevant to this routine are given in volume II, section 6 (table 6-1), of this user guide.

The SETUP5 subprogram also reads in the MSS DATAPE.

6.2.4 OUTPUTS

The results are returned for use by the calling routine.

6.2.5 STORAGE REQUIREMENTS

This subprogram requires 4326 bytes of storage.

6.2.6 DESCRIPTION

Processor control card images and an MSS DATAPE are input to the SETUP5 subprogram. The reread buffer is set up, and the CHANNELS, HED1, HED2, COMMENT, DATE, DISPLAY, SIZE, and DATAFILE control card images are read into the program. Subprogram SORT is called to arrange the CHANNEL numbers in ascending order. The DATAFILE entry is checked for errors, and if an error is detected a message is generated. SETUP5 ascertains that the DISPLAY card image is a subset of the CHANNELS card image, checks the minimum and maximum x-values to be histogrammed, establishes bases for ARRAY, resets the number of channels to be displayed, recalculates address TOTTL, zeros out channels that are not to be plotted, and generates appropriate messages in the event of errors.

6.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 6.4.

6.2.8 LISTING

The subprogram listing is provided in volume IV, section 6.

C-2

6.3 SORT

The SORT subprogram arranges a vector of integers in ascending order.

6.3.1 LINKAGES

This routine does not call any other subprogram. It is called by SETUP5.

6.3.2 INTERFACES

The SORT subprogram interfaces with other routines through the calling arguments.

6.3.3 INPUTS

Calling sequence: CALL SORT(IA,IPT,NUMVEG,IPTVEC)

Parameter	Dimension	In/out	Definition
IA	1	In	Number of elements to be sorted.
IPT	2	Out	Array containing beginning and ending pointer for IPTVEC.
NUMVEC	30	In	Array containing elements to be sorted.
IPTVEC	30	Out	Array containing pointers for NUMVEC.

6.3.4 OUTPUTS

The results are returned for use by the calling routine.

6.3.5 STORAGE REQUIREMENTS

This subprogram requires 884 bytes of storage.

6.3.6 DESCRIPTION

Not required.

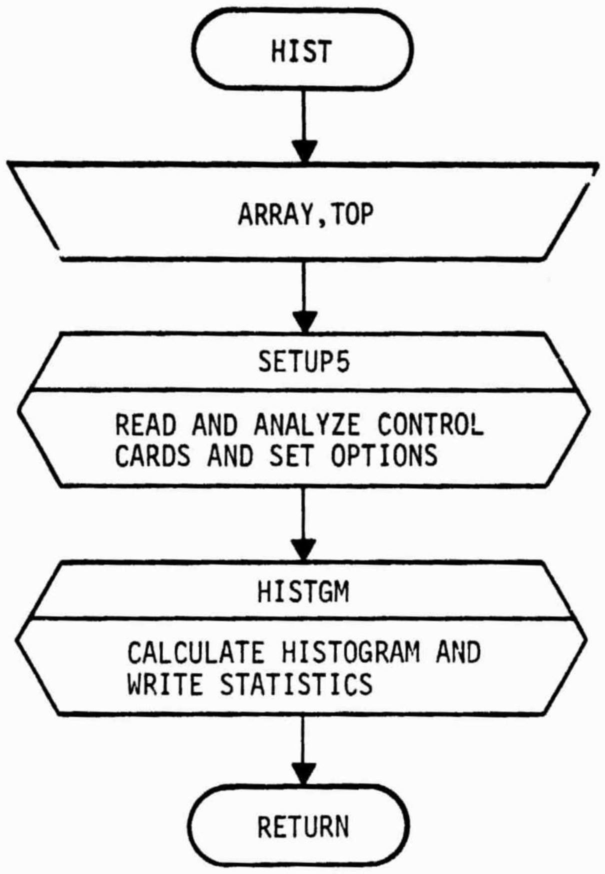
6.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 6.4.

6.3.8 LISTING

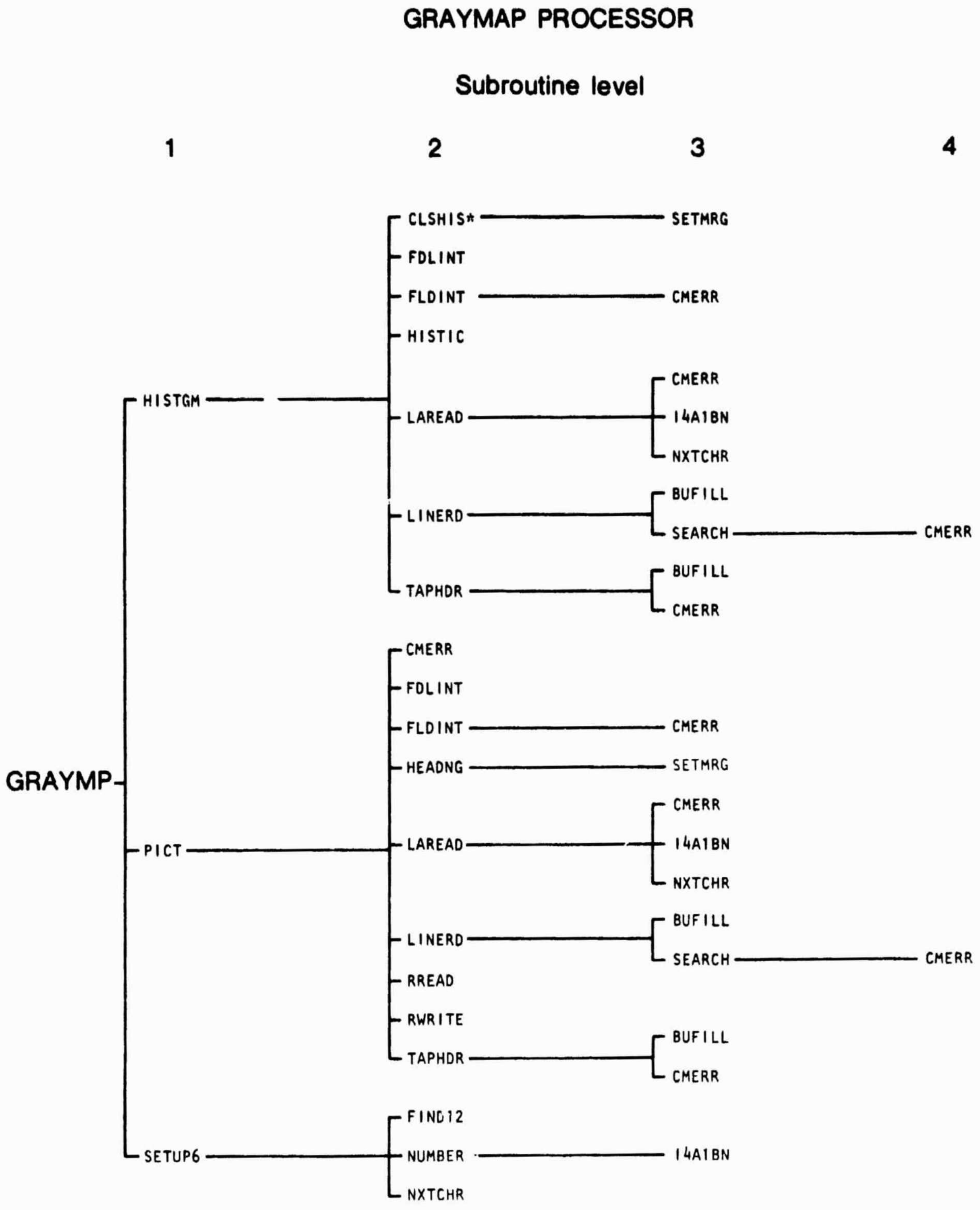
The subprogram listing is provided in volume IV, section 6.

6.4 SUBPROGRAM FLOW CHARTS


HIST
ARRAY,TOP
SETUP5
READ AND ANALYZE CONTROL CARDS AND SET OPTIONS
HISTGM
CALCULATE HISTOGRAM AND WRITE STATISTICS
RETURN

7. GRAYMAP PROCESSOR SUBPROGRAMS

The GRAYMAP processor produces alphanumeric pictorial printouts of remotely sensed, digitized MSS image data. Only four programs are exclusive to the GRAYMAP processor: the driver routine GRAYMP; SETUP6, which reads in control card images; HEADNG, which prints the heading for the pictorial printout; and PICT, which displays the pictorial features. The utility subprogram HISTGM is called to histogram data, if required. (The utility subprograms are described in section 19.) Figure 7-1 is a linkage diagram for the GRAYMAP processor.



*Entry points are COMHST, FLDHIS, and HSTGRM.

Figure 7-1.- Linkage diagram for the GRAYMAP processor.

7.1 GRAYMP

The GRAYMP subprogram is the driver routine for the GRAYMAP processor.

7.1.1 LINKAGES

The GRAYMP routine calls the HISTGM, PICT, and SETUP6 subprograms. It is called by MO.‹TOR.

7.1.2 INTERFACES

The GRAYMP subprogram interfaces with other routines through common blocks GLOBAL, GRCBLK, and HISTOR and through the calling arguments.

7.1.3 INPUTS

Calling sequence: CALL GRAYMP(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

7.1.4 OUTPUTS

Not applicable.

7.1.5 STORAGE REQUIREMENTS

This subprogram requires 2778 bytes of storage.

7.1.6 DESCRIPTION

The GRAYMP subprogram calls SETUP6, which accepts tape input and reads control card images; HISTGM, which performs histograms and outputs statistics; and PICT, which pictorially displays the requested features.

7.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 7.5.

7.1.8 LISTING

The subprogram listing is provided in volume IV, section 7.

7.2 HEADNG

The HEADNG subprogram prints out the heading for the pictorial display of histogrammed data.

7.2.1 LINKAGES

The HEADNG subprogram calls the SETMRG subprogram. It is called by PICT.

7.2.2 INTERFACES

The HEADNG subprogram interfaces with other routines through common blocks GLOBAL and GRCBLK and through the calling arguments.

7.2.3 INPUTS

Calling sequence: CALL HEADNG(TYPE,FETNUM,BINLEV,NUMBIN,FLDINP, SYMBOL,NSAMP,FET,SYMDIM,TCOL)

Parameter	Dimension	In/out	Definition
TYPE	1	In	Column heading.
FETNUM	1	In	Number of features (channels) in FETVC2.
BINLEV	30,16	In	Array of bin levels (≤16) for each channel (≤30).
NUMBIN	1	In	Number of bins in BINLEV.
FLDINP	7	In	Array of fields for which gray-scale maps are requested.
SYMBOL	16,2	In	Array of possible symbols for the 16 gray-level assignments of data values.
NSAMP	1	In	Sample numbers.

Parameter	Dimension	In/out	Definition
FET	1	In	Array of features (channels) to graymap.
SYMDIM	1	In	Column number in the symbol table.
TCOL	1	In	Column number.

7.2.4 OUTPUTS

This subprogram prints results on the specified output unit.

7.2.5 STORAGE REQUIREMENTS

This subprogram requires 2916 bytes of storage.

7.2.6 DESCRIPTION

Not applicable.

7.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 7.5.

7.2.8 LISTING

The subprogram listing is provided in volume IV, section 7.

7.3 PICT

The PICT subprogram pictorially displays the requested features.

7.3.1 LINKAGES

This routine calls the CMERR, FDLINT, FLDINT, HEADNG, LAREAD, LINERD, RREAD, RWRITE, and TAPHDR subprograms. PICT is called by GRAYMP.

7.3.2 INTERFACES

The PICT subprogram interfaces with other routines through common blocks GLOBAL and GRCBLK and through the calling arguments.

7.3.3 INPUTS

Input to the PICT subprogram consists of the MSS DATAPE and the HISFIL tape output by the HIST processor.

Calling sequence: CALL PICT(IDATA)

Parameter	Dimension	In/out	Definition
IDATA	12 000	In	Array of scanner data.

7.3.4 INPUTS

This subprogram outputs graymapped features on the specified unit.

The results are returned for use by the calling routine.

7.3.5 STORAGE REQUIREMENTS

This subprogram requires 54 920 bytes of storage.

7.3.6 DESCRIPTION

The PICT subprogram calls TAPHDR and LAREAD to read in data. If the data will not fit on the storage drum, it reduces the number of channels by 1 until the data are the appropriate size and instructs the user to run GRAYMP again to process the channels that were not input. It calls LINERD to read in data one line at a time and RWRITE to write each line of data. For each channel, it:

a. Writes the channel, field name, sample and line increments, and vertices.

b. Sets up a value-symbol table for the graymap.

c. Initializes a tape read for the sample data.

d. Calls HEADNG to print the heading for the pictorial display.

e. Calls RREAD to read and fill 20 buffers.

f. Writes each line of data for the graymap symbol display.

Reads in another field definition card via LAREAD and repeats this procedure until all input fields have been displayed.

7.3.7 FLOW CHART

The available flow charts for this processor are provided in section 7.5.

7.3.8 LISTING

The subprogram listing is provided in volume IV, section 7.

7.4 SETUP6

The SETUP6 subprogram reads and analyzes control card images and sets options for the GRAYMAP processor.

7.4.1 LINKAGES

The SETUP6 subprogram calls the FIND12, NUMBER, and NXTCHR subprograms. It is called by the GRAYMP driver routine.

7.4.2 INTERFACES

The SETUP6 subprogram interfaces with other routines through common blocks GLOBAL, GRCBLK, and HISTOR and through the calling arguments.

7.4.3 INPUTS

Calling sequence: CALL SETUP6(FILHIS,BINCNT,BINLEV,NUMBIN,SYMBOL, SYMCNT,SYMDIM)

Parameter	Dimension	In/out	Definition
FILHIS	NOFEAT,256	In	Array containing histogram data.
BINCNT	1	In/out	Number of channels in BINLEV ($\leq$30).
BINLEV	30,16	Out	Array of bin levels ($\leq$16) for each channel ($\leq$30).
NUMBIN	1	Out	Numbers of bins in BINLEV ($\leq$16).
SYMBOL	16,2	Out	Any of 16 possible symbols for gray level assignments of data values.
SYMCNT	1	Out	Row number in the symbol table.
SYMDIM	1	Out	Column number in the symbol table.

The control cards relevant to this routine are given in volume II, section 7 (table 7-1), of this user guide.

7.4.4 OUTPUTS

This subprogram outputs supervisor information and channels and symbols to be used in processing.

Supervisor information is returned for use by the calling routine.

7.4.5 STORAGE REQUIREMENTS

This subprogram requires 6080 bytes of storage.

7.4.6 DESCRIPTION

The subprogram SETUP6 sets up the reread buffer to read in all processor control card images. The CHANNELS, BINLEVEL, and SYMBOL card images are checked to assure that they are not out of range. The DATAFILE positioning card is read and checked for errors; if an error is present, a message is generated. SETUP6 then reads the *END card image, lists default symbols, and checks for input bin levels. It reads the histogram, calculates bin levels (if not input), checks to see if channels have been histogrammed, and computes bin levels for each channel. It prints the supervisor information, the channels to be graymapped, and the symbols to be used.

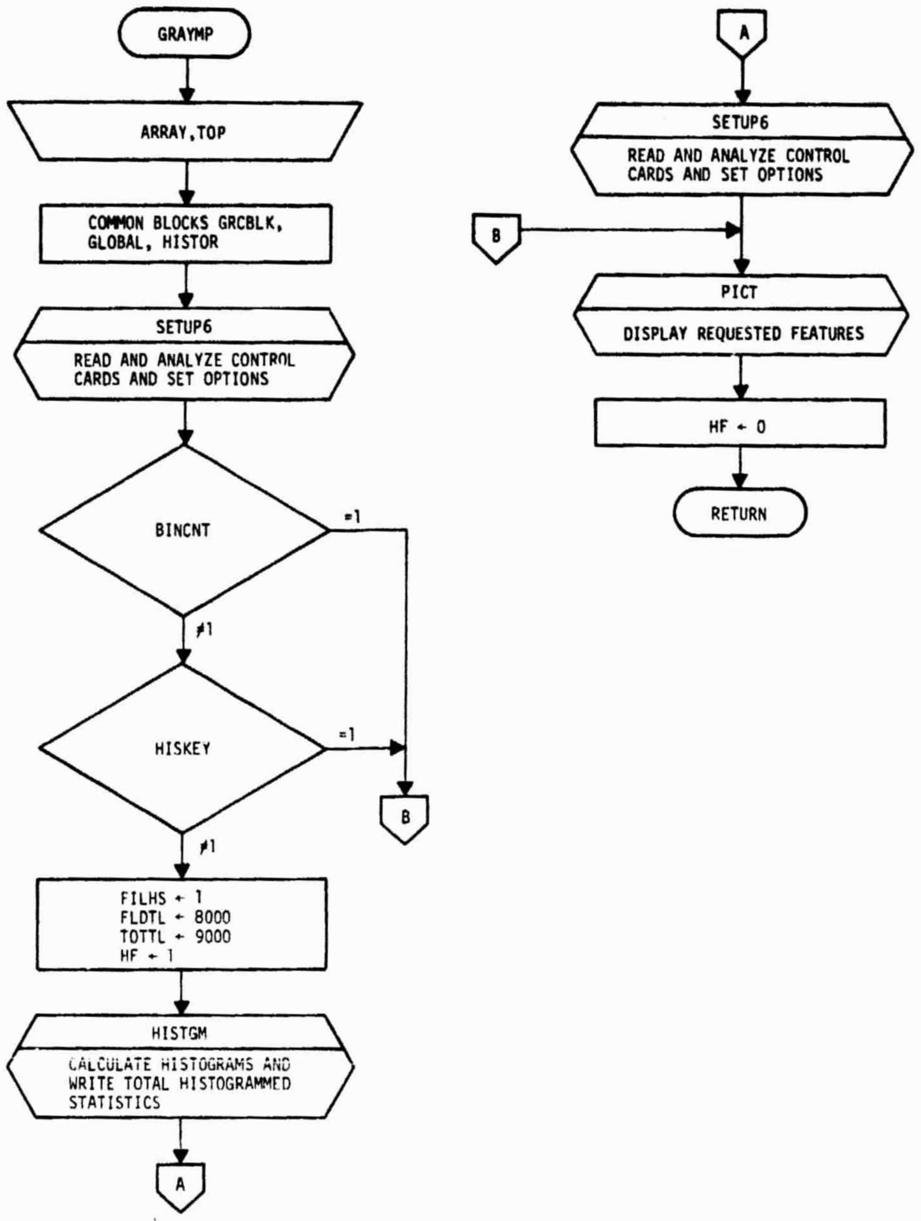
7.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 7.5.

7.4.8 LISTING

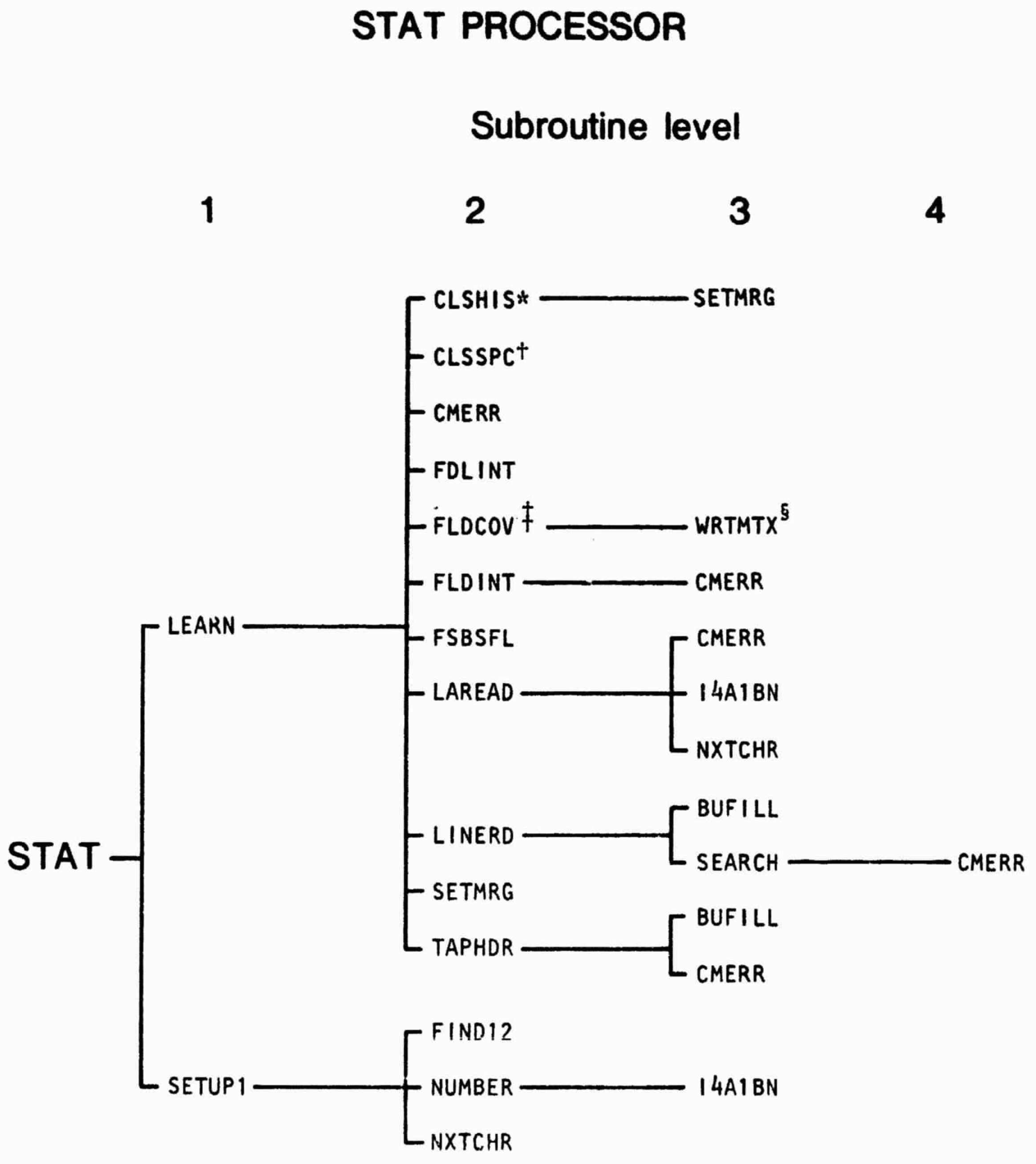
The subprogram listing is provided in volume IV, section 7.

7.5 SUBPROGRAM FLOW CHARTS


GRAYMP
ARRAY,TOP
COMMON BLOCKS GRCBLK, GLOBAL, HISTOR
SETUP6
READ AND ANALYZE CONTROL CARDS AND SET OPTIONS
BINCNT
=1
≠1
HISKEY
=1
B
≠1
FILHS ← 1
FLDTL ← 8000
TOTTL ← 9000
HF ← 1
HISTGM
CALCULATE HISTOGRAMS AND WRITE TOTAL HISTOGRAMMED STATISTICS
A
A
SETUP6
READ AND ANALYZE CONTROL CARDS AND SET OPTIONS
B
PICT
DISPLAY REQUESTED FEATURES
HF ← 0
RETURN

8. STAT PROCESSOR SUBPROGRAMS

The STAT processor computes multichannel means, standard deviations, covariance matrices, and correlation coefficients for each training field and all training subclasses defined through user input. If requested by the user, histograms and spectral plots are also produced. Card images and raw multispectral data are input. The STAT driver routine calls the SETUP1 subprogram to read and analyze control card images and the LEARN subprogram to process raw data, compute statistics, and output a SAVTAP file. LEARN calls two subprograms that are exclusive to the processor, CLSSPC and FLDCOV. Both SETUP1 and LEARN call a number of utility subprograms (documented in section 19). Figure 8-1 is a linkage diagram for the STAT processor.



*Entry points are COMHST, FLDHIS, and HSTGRM.

†Entry points are FLDSPC and MULSPC.

‡Entry point is CLSCOV.

§Entry point is DWRTMX.

Figure 8-1.- Linkage diagram for the STAT processor.

8.1 STAT

The STAT subprogram is the driver routine for the STAT processor.

8.1.1 LINKAGES

The STAT routine calls the LEARN and SETUP1 subprograms. It is called by MONTOR.

8.1.2 INTERFACES

The STAT subprogram interfaces with other routines through common blocks STBASE and STCBLK and through the calling arguments.

8.1.3 INPUTS

Calling sequence: CALL STAT(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1500*	In/out	Double-precision working storage. ARRAY is a block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

8.1.4 OUTPUTS

Not applicable.

8.1.5 STORAGE REQUIREMENTS

This subprogram requires 1048 bytes of storage.

*Double precision.

8.1.6 DESCRIPTION

The STAT routine coordinates the various subprograms to obtain the multichannel means, standard deviations, covariance matrices, and correlation coefficients for user-defined training fields and subclasses. It calls SETUP1 to read and analyze control cards and LEARN to build the SAVTAP file.

8.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 8.6.

8.1.8 LISTING

The subprogram listing is provided in volume IV, section 8.

8.2 CLSSPC

The CLSSPC subprogram generates a spectral plot for each user-defined training field and/or subclass. CLSSPC has two additional entry points: FLDSPC, which plots the spectral responses for the training fields; and MULSPC, which plots the multispectral responses for all training subclasses.

8.2.1 LINKAGES

The CLSSPC subprogram does not call any other subprogram. It is called by the LEARN subprogram.

8.2.2 INTERFACES

The CLSSPC subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

8.2.3 INPUTS

Calling sequences:

a. CALL CLSSPC(MEAN,SUBSTD,IDVEC,PTRVEC,PLOT,TITLE,NOFEAT, FETVEC,SPCBAS)

Parameter	Dimension	In/out	Definition
MEAN	1	Out	Array of subclass means.
SUBSTD	1	Out	Array of subclass standard deviations.
IDVEC	1	In	Array of training field information.
PTRVEC	5	In	Vector of pointers used in computing base address of channel means (MENBAS).

Parameter	Dimension	In/out	Definition
PLOT	4,NOFEAT,49	Out	Array for output of the spectral plot.
TITLE	1	In	Name of the training subclass being plotted.
NOFEAT	1	In	Number of features (channels).
FETVEC	30	In	Vector of channel numbers.
SPCBAS	1	In	Minimum radiance value of spectral plot on y-axis.

b. ENTRY FLDSPC(DMEAN,DEV,IDVEC,PTRVEC,PLOT,MEAN,SUBSTD,FLDNAM, NOFEAT,FETVEC,SPCBAS)

Parameter	Dimension	In/out	Definition
DMEAN	1*	In	Array of field means.
DEV	1*	In	Array of standard deviations.
FLDNAM	1	In	Name of the training field being plotted.

c. ENTRY MULSPC(MEAN,SUBSTD,JDVEC,PTRVEC,PLOT,NOFEAT,FETVEC, SPCBAS)

Parameter	Dimension	In/out	Definition
JDVEC	1,1	In	Array of subclass names.

8.2.4 OUTPUTS

This subprogram outputs a spectral or multispectral plot on the specified unit.

*Double precision.

8.2.5 STORAGE REQUIREMENTS

This subprogram requires 4086 bytes of storage.

8.2.6 DESCRIPTION

At the user's option, the CLSSPC subprogram accepts statistics, radiance values, and field information as input and plots and prints a matrix of spectral values for each user-defined training field and subclass.

8.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 8.6.

8.2.8 LISTING

The subprogram listing is provided in volume IV, section 8.

8.3 FLDCOV

The FLDCOV subprogram calculates the covariance matrices and correlation coefficients for the fields. An additional entry point, CLSCOV, is used to calculate the same statistics for the training subclasses.

8.3.1 LINKAGES

The FLDCOV subprogram calls the WRTMTX subprogram. It is called by LEARN.

8.3.2 INTERFACES

The FLDCOV subprogram interfaces with other routines through common block GLOBAL and through the calling arguments.

8.3.3 INPUTS

Calling sequences:

a. CALL FLDCOV(COR,DEV,MEAN,VAR,PTS,GO,FLDNAM,NOFEAT,MAXFET,
   VARSIZ)

Parameter	Dimension	In/out	Definition
COR	VARSIZ*	Out	Array containing correlation coefficients.
DEV	NOFEAT*	Out	Array containing standard deviations.
MEAN	NOFEAT*	In/out	Array containing means.
VAR	VARSIZ*	In/out	Array of covariance matrices.
PTS	1	In	Number of points in the field.
GO	1	In	Switch used to determine if a new page heading is needed.

*Double precision.

Parameter	Dimension	In/out	Definition
FLDNAM	1	In	Field name.
NOFEAT	1	In	Number of features (channels).
MAXFET	1	In	Maximum number of channels.
VARSIZ	1	In/out	Size of each covariance matrix: $\mathrm{NOFEAT}\left(\frac{\mathrm{NOFEAT}\ +\ 1}{2}\right)$.

b. ENTRY CLSCOV(COR,DEV,MEAN,VAR,PTS,GO,TITLE,NOFEAT,MAXFET, VARSIZ)

Parameter	Dimension	In/out	Definition
TITLE	1	In	Field name.

8.3.4 OUTPUTS

This subprogram outputs the covariance matrix and correlatior coefficients on the specified unit.

8.3.5 STORAGE REQUIREMENTS

This subprogram requires 2666 bytes of storage.

8.3.6 DESCRIPTION

The FLDCOV subprogram uses three equations to calculate statistics:

$$\mathrm{COVAR}(1,2) = \frac{1}{N - 1}\left[\sum_{J=1}^{N} X_1X_2 - U_1U_2\right] \tag{8-1}$$

$$\mathrm{MEAN}(1) = \frac{1}{N}\sum_{I=1}^{N} X_1 = U_1 \tag{8-2}$$

$$\mathrm{STDEV}(2) = \sqrt{\mathrm{COVAR}(2,2)} \tag{8-3}$$

where

N = number of channels

X = data point

U = estimate of mean

The WRTMTX subprogram (entry DWRTMX) is called to print results.

8.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 8.6.

8.3.8 LISTING

The subprogram listing is provided in volume IV, section 8.

8.4 LEARN

The LEARN subprogram computes the multichannel means, standard deviations, covariance matrix, and correlation coefficient for each training field and all training subclasses which are defined through user input.

8.4.1 LINKAGES

The LEARN subprogram calls the CLSHIS, CLSSPC, CMERR, FDLINT, FLDCOV, FLDINT, FSBSFL, LAREAD, LINERD, SETMRG, and TAPHDR subprograms. It is called by the STAT driver routine.

8.4.2 INTERFACES

The LEARN subprogram interfaces with other routines through common blocks GLOBAL, STBASE, and STCBLK and through the calling arguments.

8.4.3 INPUTS

Input to the LEARN subprogram consists of the MSS DATAPE and field definition cards.

Calling sequence: CALL LEARN(SPEC,COVAR,AVAR,CLSDES,SUBSAV, FLDMEN,FLDVAR,SUBMEN,SUBVAR,SUBSTD,SUBCLS,HFTALY,HSTALY,FLDSAV, SAVERT,KEPPTS,MAXSUB)

Parameter	Dimension	In/out	Definition
SPEC	5,NOSPEC	In/out	Array of spectral information (radiance values).
COVAR	VARSIZ	Out	Array of field covariances.
AVAR	NOFEAT,MAXSUB	Out	Array of field means.
CLSDES	MAXSUB	Out	Array of class names.

Parameter	Dimension	In/out	Definition
SUBSAV	4,MAXSUB	Out	Array of subclass statistics.
FLDMEN	NOFEAT*	Out	Array of field means.
FLDVAR	VARSIZ*	Out	Array of variances.
SUBMEN	NOFEAT*	Out	Array of subclass means.
SUBVAR	VARSIZ*	Out	Array of subclass variances.
SUBSTD	NOFEAT,MAXSUB	Out	Array of subclass standard deviations.
SUBCLS	1	Out	Array of subclass numbers.
HFTALY	NOHIST,XSIZ	Out	Array of field histograms.
HSTALY	NOHIST,XSIZ	Out	Array of subclass histogram totals.
FLDSAV	10,MAXFLD	Out	Array of field information for each field index: 1 = field name 2 = class number 3 = subclass number 4 = number of vertices 5 = starting line number 6 = ending line number 7 = starting sample number 8 = ending sample number 9 = line increment 10 = sample increment
SAVERT	22,MAXFLD	Out	Array of training field vertices.
KEPPTS	MAXSUB	Out	Array of pixels in each subclass.
MAXSUB	1	In	Maximum number of subclasses.

*Double precision.

8.4.4 OUTPUTS

This subprogram outputs statistics on the SAVTAP file.

8.4.5 STORAGE REQUIREMENTS

This subprogram requires 61 924 bytes of storage.

8.4.6 DESCRIPTION

Raw data from the MSS DATAPE and field definition card images are input to the LEARN subprogram. LEARN invokes the utility subprogram CLSHIS to produce histograms and processor subprograms CLSSPC to produce spectral plots and FLDCOV to calculate the covariance matrices and correlation coefficients. Various utility subprograms are called, also, to read in or to output data. The LEARN subprogram plots spectral responses first for each field and then for each subclass; calculates the covariance matrix and mean vector for each field and subclass; and saves the subclass mean, covariance, and standard deviation in the arrays AVAR, COVAR, and SUBSTD. These statistics are output on the SAVTAP file. Multispectral plots are stored in the array SPEC, and a composite spectral plot of all training subclasses is produced.

8.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 8.6.

8.4.8 LISTING

The subprogram listing is provided in volume IV, section 8.

8.5 SETUP1

The SETUP1 subprogram reads and analyzes control card images and sets options for the STAT processor.

8.5.1 LINKAGES

The SETUP1 subprogram calls the FIND12, NUMBER, and NXTCHR subprograms. It is called by the STAT driver routine.

8.5.2 INTERFACES

The SETUP1 subprogram interfaces with other routines through common blocks GLOBAL, STBASE, and STCBLK and through the calling arguments.

8.5.3 INPUTS

Calling sequence: CALL SETUP1(SPCVEC,TOP,MAXSUB)

Parameter	Dimension	In/out	Definition
SPCVEC	5,20	In	Array of radiance values to be plotted.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
MAXSUB	1	In	Maximum number of subclasses.

The control cards relevant to this routine are given in volume II, section 8 (table 8-1) of this user guide.

8.5.4 OUTPUTS

This subprogram outputs a list of user-requested options.

8.5.5 STORAGE REQUIREMENTS

This subprogram requires 9654 bytes of storage.

8.5.6 DESCRIPTION

The SETUP1 subprogram sets up the reread buffer and reads the following control card images, one at a time:

- OPTION – If an error is detected on an OPTION card, the read is discontinued and an error message is generated.
- CHANNELS – Out-of-range requested features are eliminated, and the resulting channel vector is arranged.
- HISTOGRAM – Out-of-range requested subclasses are eliminated, and the resulting subclass vector is arranged.
- SPEC – Read to determine the number of groups of subclasses specified (≤ 20) for which spectral plots will be generated.
- IBLOCK – Read in and used only if an alternate method of setting options is desired.
- SIZE – Read to determine the value of the minimum radiance value on the y-axis of the spectral plot.
- DATE, COMENT, HED1, and HED2 cards – Read in to enter the date of execution, comments, and headings.
- DATAFILE and STATFILE cards – Read in and, if errors occur, messages are generated.

SETUP1 calculates the bases of processor arrays and prints out user-requested options. Execution terminates with the appropriate message if an erroneous control card image is encountered, if excessive options are requested, or if channels or subclasses are excessive.

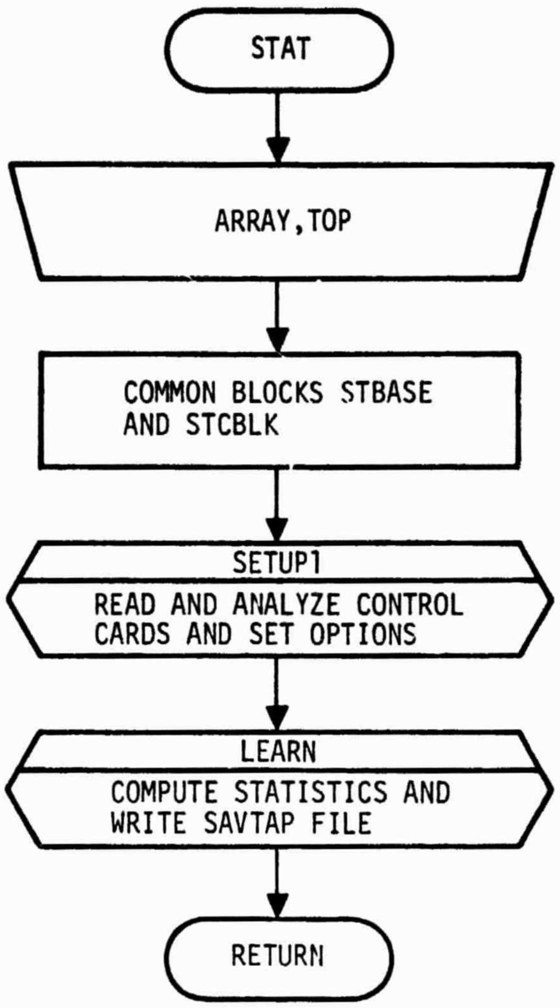
8.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 8.6.

8.5.8 LISTING

The subprogram listing is provided in volume IV, section 8.

8.6 SUBPROGRAM FLOW CHARTS


STAT
ARRAY,TOP
COMMON BLOCKS STBASE AND STCBLK
SETUP1
READ AND ANALYZE CONTROL CARDS AND SET OPTIONS
LEARN
COMPUTE STATISTICS AND WRITE SAVTAP FILE
RETURN

9. ISOCLS PROCESSOR SUBPROGRAMS

The ISOCLS processor performs a modified version of the clustering algorithm originally developed at Stanford Research Institute (ref. 1). The original version was modified (ref. 2) and resulted in the present ISOCLS processor, which utilizes 5 processor subprograms and 40 utility subprograms (documented in section 19). Figure 9-1 is a linkage diagram for the ISOCLS processor.

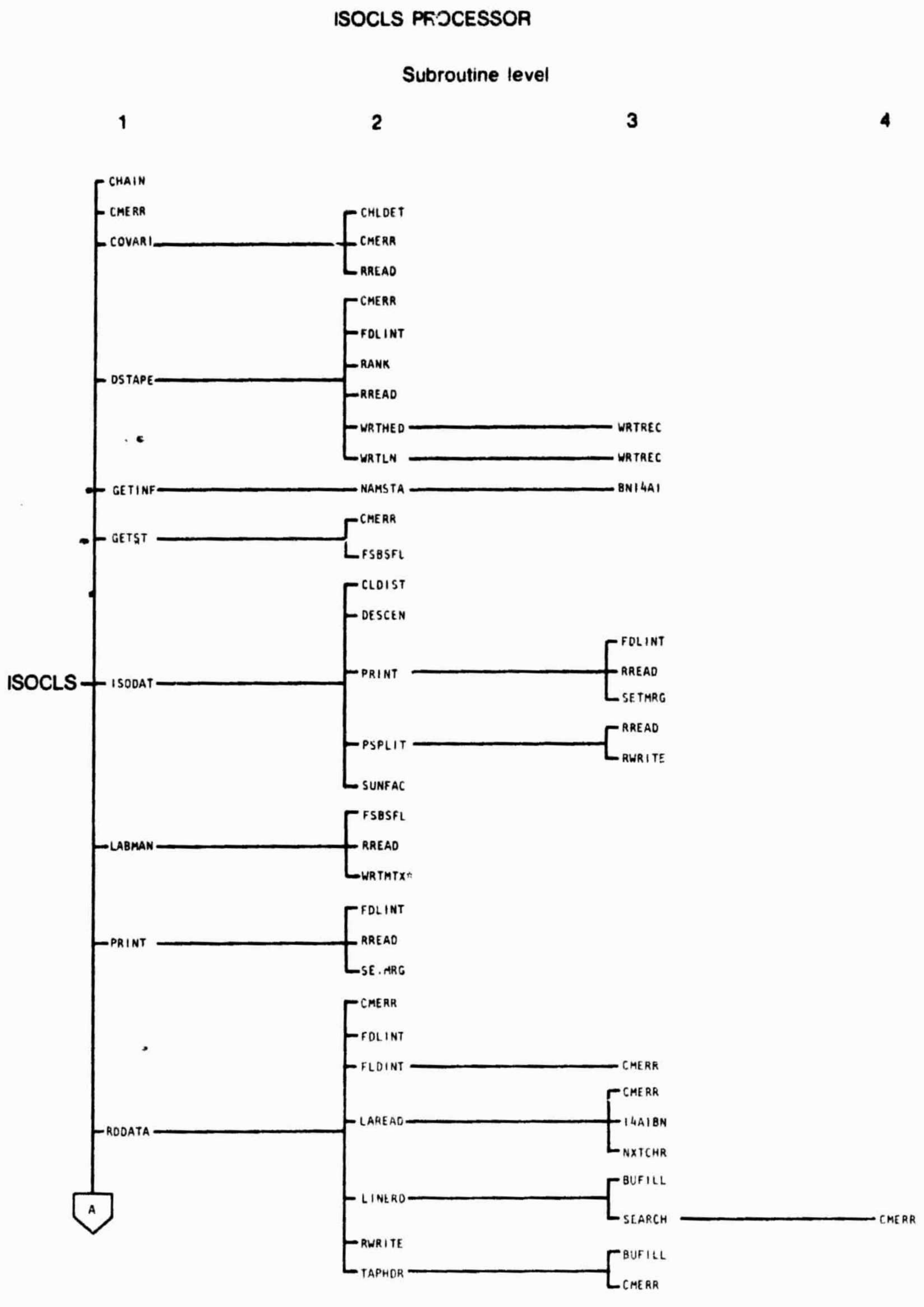
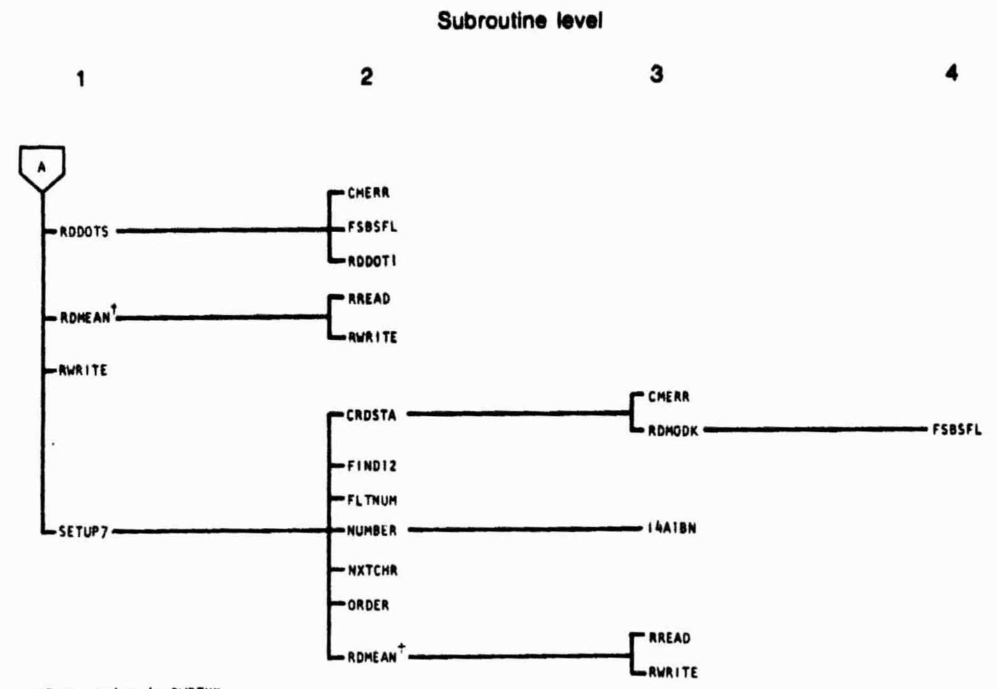


Figure 9-1.- Linkage diagram for the ISOCLS processor.



*Entry point is DWRTMX.
†Entry point is RDFILE.

Figure 9-1.- Concluded.

9.1 ISOCLS

The ISOCLS subprogram is the driver routine for the ISOCLS processor.

9.1.1 LINKAGES

The ISOCLS routine calls the CHAIN, CMERR, COVAR1, DSTAPE, GETINF, GETST, ISODAT, LABMAN, PRINT, RDDATA, RDDOTS, RDMEAN, RWRITE, and SETUP7 subprograms. It is called by MONTOR.

9.1.2 INTERFACES

The ISOCLS subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, and PASS and through the calling arguments.

9.1.3 INPUTS

Calling sequence: CALL ISOCLS(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY; TOP = 10 600.

The control cards relevant to this routine are given in volume II, section 9 (table 9-1), of this user guide.

9.1.4 OUTPUTS

Not applicable.

9.1.5 STORAGE REQUIREMENTS

This subprogram requires 52 222 bytes of storage.

9.1.6 DESCRIPTION

The ISOCLS subprogram uses the random-access disk file as four distinct files:

- IBEGIN – Beginning address for input initial cluster centers.
- BEGIN1 – Beginning address for image data.
- BEGIN2 – Beginning address for IPLACE (cluster to which corresponding point belongs).
- BEGIN3 – Beginning address for temporary storage of class statistics.

The ISOCLS subprogram reserves enough disk storage for maximum initial means and utilizes a number of processor and utility subprograms to perform the following functions:

- Read card input and initialize default values.
- Coordinate reading of data.
- Perform clustering.
- Link clusters that are within a specified distance of one another.
- Output final clustering results.
- Create an output file on the MAPUNT (optional).
- Calculate the covariance matrix for each cluster.
- Store statistics.
- Write a SAVTAP file.

9.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 9.6.

9.1.8 LISTING

The subprogram listing is provided in volume IV, section 9.

9.2 COVAR1

The COVAR1 subprogram calculates and prints the covariance matrix for each cluster.

9.2.1 LINKAGES

The COVAR1 subprogram calls the CHLDET, CMERR, and RREAD subprograms. It is called by the ISOCLS driver routine.

9.2.2 INTERFACES

The COVAR1 subprogram interfaces with other routines through common blocks GLOBAL and PASS and through the calling arguments.

9.2.3 INPUTS

Calling sequence: CALL COVAR1(COVAR,C,IPLACE,MEANS,N,IBAD)

Parameter	Dimension	In/out	Definition
COVAR	VARSIZ,LNCAT	Out	Array containing covariance matrix for each cluster; VARSIZ = size of each covariance matrix; LNCAT = number of clusters.
C	NOFEAT,NOPTS	In	Array containing pixel radiance values to be clustered; NOPTS = number of points in each record.
IPLACE	NOPTS	In	Vector of cluster numbers to which corresponding data points (pixels) are assigned.
MEANS	NOFEAT,MAXCLS	In	Array containing means of each feature for each cluster.

Parameter	Dimension	In/out	Definition
N	MAXCLS	In	Array containing field and class information for the cluster being processed.
IBAD	1	Out	Indicator that cluster was deleted because of singular covariance matrix.

9.2.4 OUTPUTS

This subprogram outputs the covariance matrix on the specified unit.

9.2.5 STORAGE REQUIREMENTS

This subprogram requires 2336 bytes of storage.

9.2.6 DESCRIPTION

Since the covariance matrix is symmetrical, only the lower triangular portion of the matrix is calculated. The elements are calculated using the following equation.

$$C_{jk} = \frac{1}{N} \sum_{1}^{N} X_j X_k - \mu_j \mu_k$$

where

C_{jk}	= the (j,k)*th* element in the covariance matrix
X_j and X_k	= input data for the j*th* and k*th* elements, respectively
μ_j and μ_k	= means of the j*th* and k*th* elements
N	= number of data points in a particular cluster

Each element is stored in the array COVAR in consecutive locations; e.g., for four features,

$$C = \begin{bmatrix} C_{11} & & & \\ C_{12} & C_{22} & & \\ C_{13} & C_{23} & C_{33} & \\ C_{14} & C_{24} & C_{34} & C_{44} \end{bmatrix}$$

CHLDET is called to compute the determinant and ascertain if the covariance matrix for each cluster is singular. If so, the cluster is deleted.

9.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 9.6.

9.2.8 LISTING

The subprogram listing is provided in volume IV, section 9.

9.3 ISODAT

The ISODAT subprogram performs the clustering process for the ISOCLS processor.

9.3.1 LINKAGES

The ISODAT subprogram calls the CLDIST, DESCEN, PRINT, PSPLIT, and SUNFAC subprograms. It is called by the ISOCLS driver routine.

9.3.2 INTERFACES

The ISODAT subprogram interfaces with other routines through common blocks GLOBAL, ISOLNK, and PASS and through the calling arguments.

9.3.3 INPUTS

Calling sequence: CALL ISODAT(C,IPLACE,MEANS,N,STDEV,CLD, FLDINF,AVP,AMN)

Parameter	Dimension	In/out	Definition
C	NOFEAT,NOPTS	In	Array containing pixel radiance values to be clustered.
IPLACE	NOPTS	Out	Vector of cluster numbers to which the corresponding data points are assigned.
MEANS	NOFEAT,MAXCLS	Out	Array containing means of each feature for each cluster.
N	MAXCLS	In	Number of pixels per cluster.
STDEV	NOFEAT,MAXCLS	In	Array containing standard deviations for each feature/ cluster.

Parameter	Dimension	In/out	Definition
CLD	MAXCLS,MAXCLS	In	Array containing distances between clusters.
FLDINF	1	In	Array containing field information.
AVP	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster standard deviations.
AMN	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster means.

9.3.4 OUTPUTS

Not applicable.

9.3.5 STORAGE REQUIREMENTS

This subprogram requires 16 304 bytes of storage.

9.3.6 DESCRIPTION

The ISODAT subprogram accepts field and statistical information as input, removes DO/DU pixels from the pixels to be clustered, applies a Sun-angle correction by channel to each pixel, and assigns data to clusters. Subprogram CLDIST is called to calculate distances between cluster centers. If the maximum number of iterations (STOP) equals zero, small clusters are deleted. If STOP does not equal zero, the program calls PRINT to output results.

ISODAT continues by deleting clusters with fewer than the minimum allowable number (NMIN) of pixels. LNCAT is reduced accordingly. The maximum standard deviation for each cluster is found, and a beginning sequence of SPLIT iterations is performed until at least 80 percent of the clusters processed have standard deviations less than the threshold parameter STDMAX. Then, iterations alternate between COMBINE and SPLIT until the last iteration (ISTOP), which is always a SPLIT iteration. Parameters are reinitialized and the iteration procedure begins again. If ISTOP is not reached, the user must initialize the next iteration by entering "CLASSIFY AND CALCULATE NEW STATISTICS." (The iteration procedure is described in detail in volume II, section 9.1.3.)

9.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 9.6.

9.3.8 LISTING

The subprogram listing is provided in volume IV, section 9.

9.4 PSPLIT

Subprogram PSPLIT performs much of the computational work associated with the ISOCLS processor. For each iteration of the clustering algorithm, it assigns pixels to clusters and computes the new mean and standard deviation vectors.

9.4.1 LINKAGES

The PSPLIT subprogram calls the RREAD and RWRITE subprograms. It is called by ISODAT.

9.4.2 INTERFACES

The PSPLIT subprogram interfaces with other routines through common blocks ISOLNK and PASS and through the calling arguments.

9.4.3 INPUTS

Calling sequence: CALL PSPLIT(MEANS,STDEV,N,CLD,C,IPLACE,AVP, AMN,MEN)

Parameter	Dimension	In/out	Definition
MEANS	NOFEAT,MAXCLS	In/out	Array containing cluster means.
STDEV	NOFEAT,MAXCLS	Out	Array containing cluster standard deviations.
N	MAXCLS	Out	Number of pixels in each cluster.
CLD	MAXCLS,MAXCLS	In	Array containing intercluster distances; not used.
C	NOFEAT,NOPTS	In/out	Array containing pixel radiance values to be clustered.

Parameter	Dimension	In/out	Definition
IPLACE	NOPTS	In/out	Vector of cluster numbers to which pixels are assigned.
AVP	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster standard deviations.
AMN	NOFEAT,MAXCLS	Out	Intermediate storage for computed cluster means.
MEN	NOFEAT,MAXCLS	Out	Not used; in ISODAT call, the same storage as MEANS.

9.4.4 OUTPUTS

The results are returned for use by the calling routine.

9.4.5 STORAGE REQUIREMENTS

This subprogram requires 3802 bytes of storage.

9.4.6 DESCRIPTION

As the pixel radiance values are read in from disk, they are first tested for 0 or 255 values over all channels. DO and DU pixels are assigned cluster numbers greater than the cluster numbers assigned to classes. Any pixel not DO or DU is assigned to a cluster based on the minimum distance L_1 from the means. At the user's option, Sun-angle correction factors can be applied to this distance computation. After all pixels are assigned to clusters, PSPLIT recomputes the cluster means and standard deviations.

In order to save calculation time, PSPLIT uses two basically identical cluster assignment loops, one which does and one which does not incorporate the Sun-angle corrections. Thus, this multiplication (with a default value of 1) is not performed for runs without Sun-angle corrections.

9.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 9.6.

9.4.8 LISTING

The subprogram listing is provided in volume IV, section 9.

9.5 RDDATA

The RDDATA subprogram coordinates the routines which read fields of data from the MSS DATAPE and stores them on disk for processing by ISOCLS.

9.5.1 LINKAGES

The RDDATA subprogram calls the CMERR, FDLINT, FLDINT, LAREAD, LINERD, RWRITE, and TAPHDR subprograms. It is called by the ISOCLS driver routine.

9.5.2 INTERFACES

The RDDATA subprogram interfaces with other routines through common blocks GLOBAL and PASS and through the calling arguments.

9.5.3 INPUTS

Input to the RDDATA subprogram consists of the MSS DATAPE and field definition card images (volume II, section 3.2.3).

Calling sequence: CALL RDDATA(ARRAY,TOP,IDATA,IDIM,LAST)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
IDATA	IDIM	In	Array for storing pixel values for each scan line.
IDIM	1	In	Maximum usable storage for IDATA.
LAST	1	Out	Flag indicating (when set to 1) that all classes for one set of control cards have been processed.

9.5.4 OUTPUTS

RDDATA outputs results on the specified unit.

9.5.5 STORAGE REQUIREMENTS

This subprogram requires 10 670 bytes of storage.

9.5.6 DESCRIPTION

The RDDATA subprogram reserves 2000 locations in ARRAY for storing field definition information. Field definition cards are read for each class and/or field and are stored as follows:

ARRAY(1) = class name

ARRAY(2) = index pointer to next class name

ARRAY(3) = number of clusters in this class

ARRAY(4) = number of fields for this class

ARRAY(5) = first field name for this class

ARRAY(6) = number of vertices for this field

ARRAY(7) = actual vertex numbers

ARRAY(8) = total pixels in this field

ARRAY(9) = field information block for this field

RDDATA reads DO and DU class name cards and the associated field definition cards, calculates field information, and computes information concerning DO/DU pixels for use by other subprograms; e.g., the number of DO/DU pixels in each field for the field being processed and the disk addresses for the DO/DU pixels. It also assigns printing symbols and special mean values (0 or 255) to DO and DU pixel radiance values. Special cluster numbers are assigned; i.e., numbers having values 1 and 2 greater than the numbers of other clusters are given to DO/DU pixels.

9.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 9.6.

9.5.8 LISTING

The subprogram listing is provided in volume IV, section 9.

9.6 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the ISOCLS processor.

10. SELECT PROCESSOR SUBPROGRAMS

The SELECT processor is the means by which the user obtains a set or linear combination of channels or features to use in processing. The user may choose one of the following criteria for evaluating the separability among subclasses.

- Weighted average interclass divergence (subprogram AVEDIV).
- Weighted average Bhattacharyya distance (subprogram BHTCHR).
- Weighted average transformed divergence (subprogram TRNDIV).

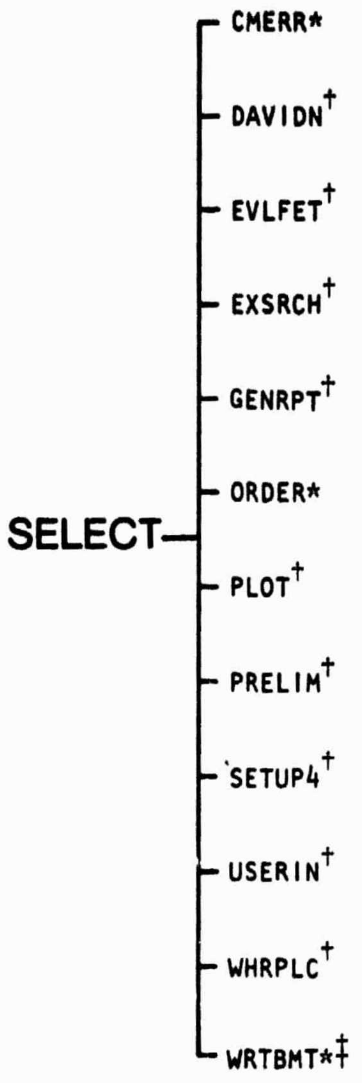
Several procedures are available for selecting the best set or a linear combination of features.

- The Davidon-Fletcher-Powell procedure (subprogram DAVIDN).
- The evaluation of user-input channels (subprogram EVLFET).
- The Exhaustive Search procedure (subprogram EXSRCH).
- The Without Replacement procedure (subprogram WHRPLC).

At the user's option, the chosen method can employ one or more of the above subprograms for measuring the separability among subclasses.

SELECT is the largest processor in the EOD-LARSYS. It has 35 subprograms that are exclusive to the processor and, in addition, calls 22 utility subprograms to assist in processing. Figure 10-1 is a linkage diagram for the SELECT processor.

SELECT PROCESSOR

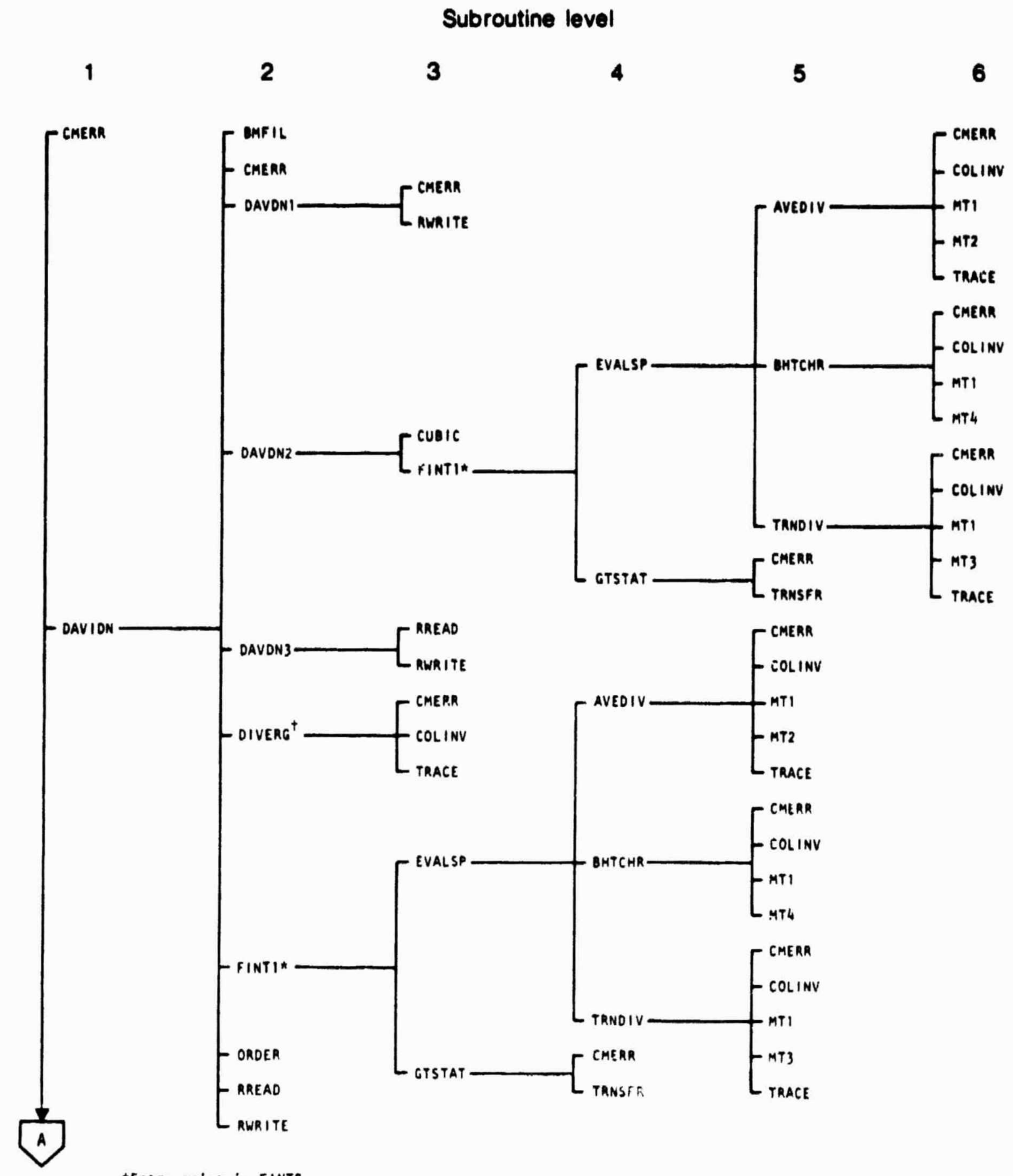


*Does not call any subroutines.

†Subroutine calls are diagrammed on succeeding pages.

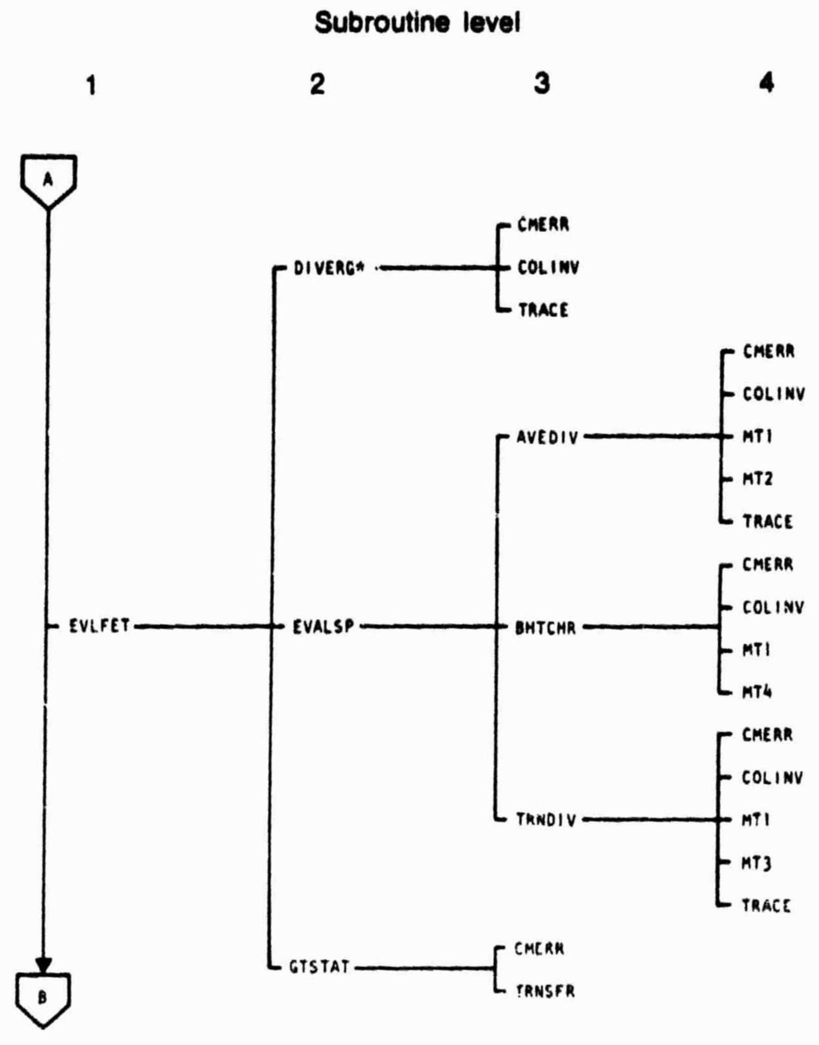
‡Entry point is WRTBM.

Figure 10-1.- Linkage diagram for the SELECT processor.



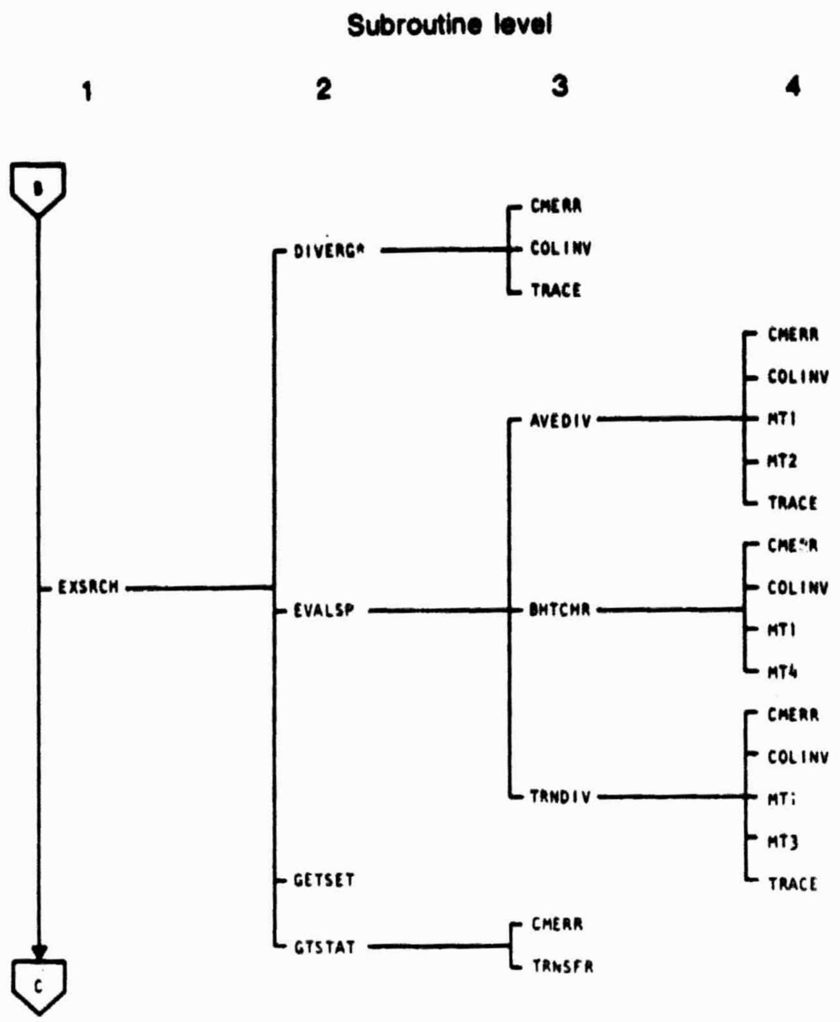
*Entry point is FINT2.
†Entry point is DIVRG1.

Figure 10-1.- Continued.



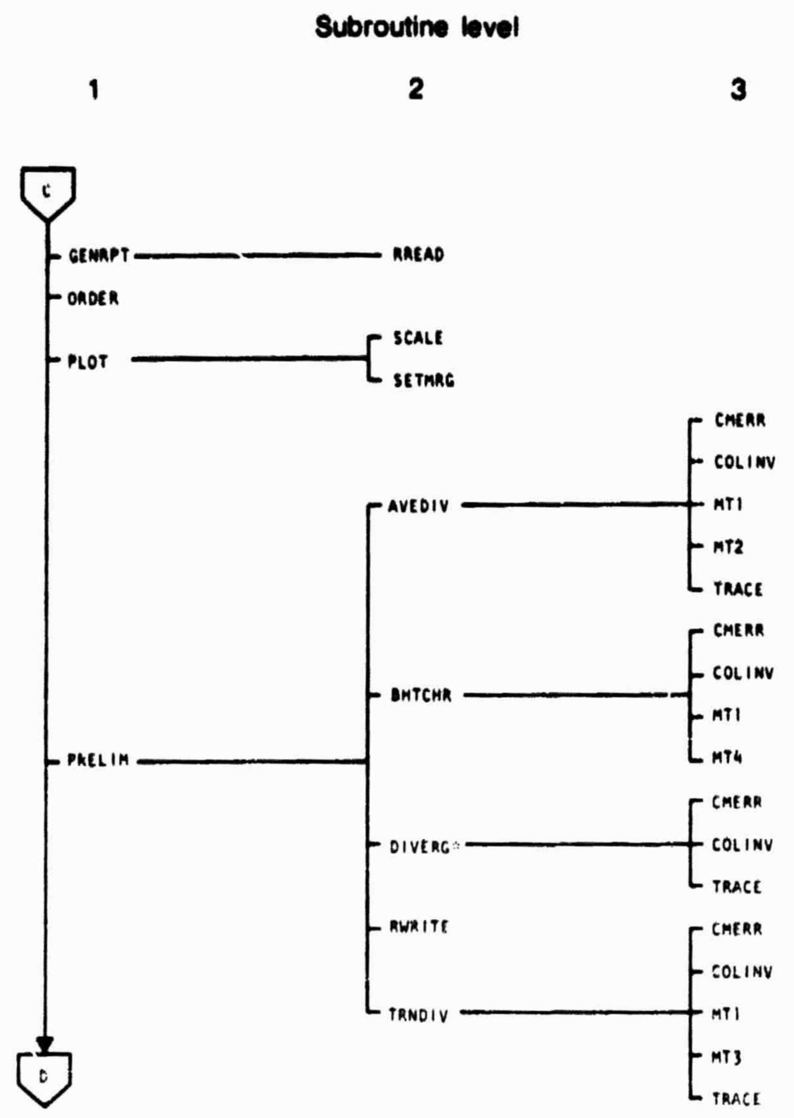
*Entry point is DIVRGI.

Figure 10-1.- Continued



*Entry point is DIVRGI.

Figure 10-1.- Continued.



*Entry point is DIVRGI.

Figure 10-1.- Continued.

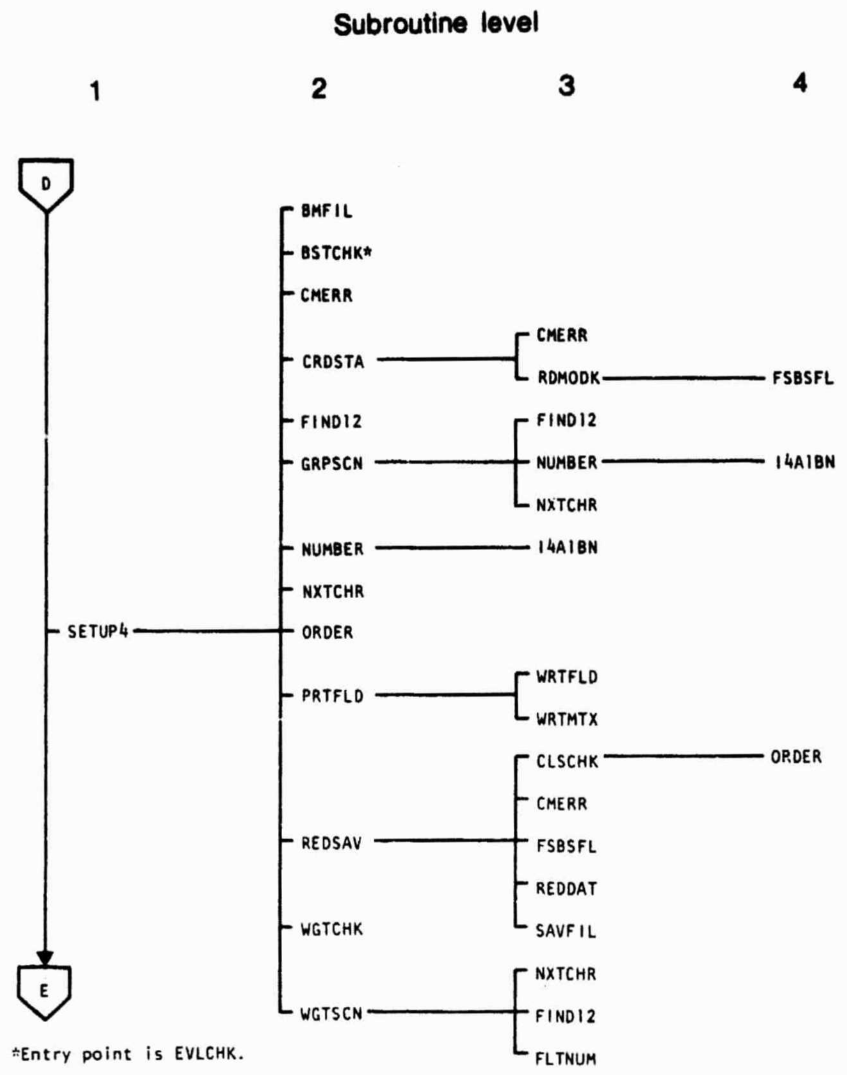
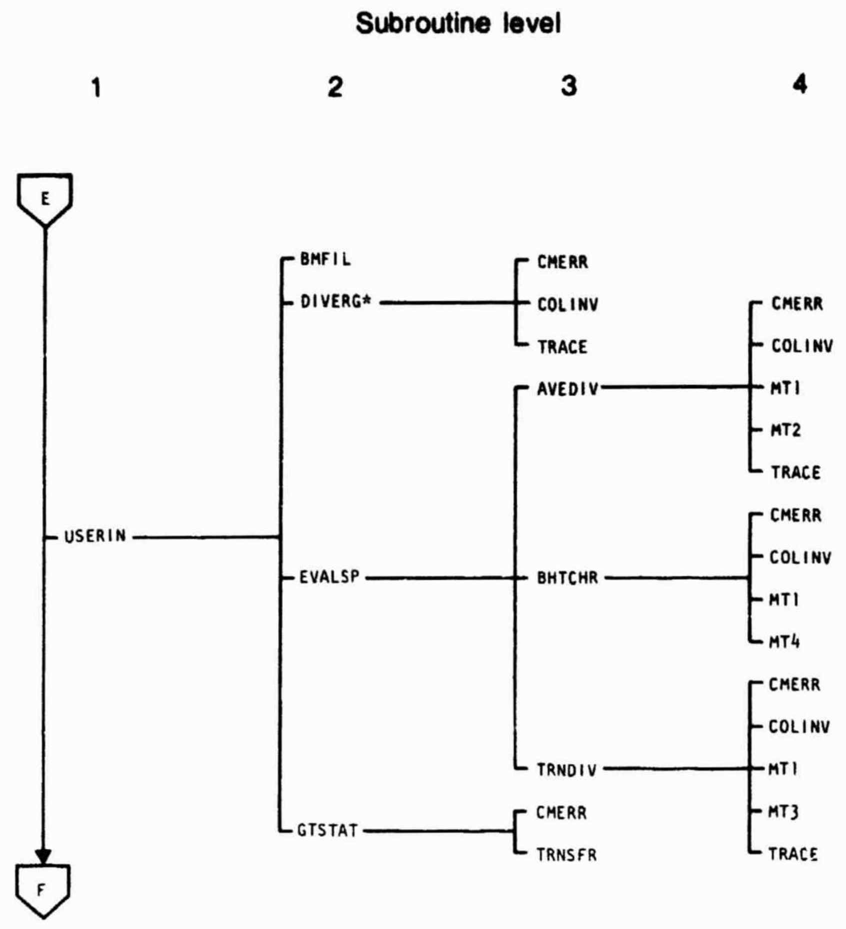
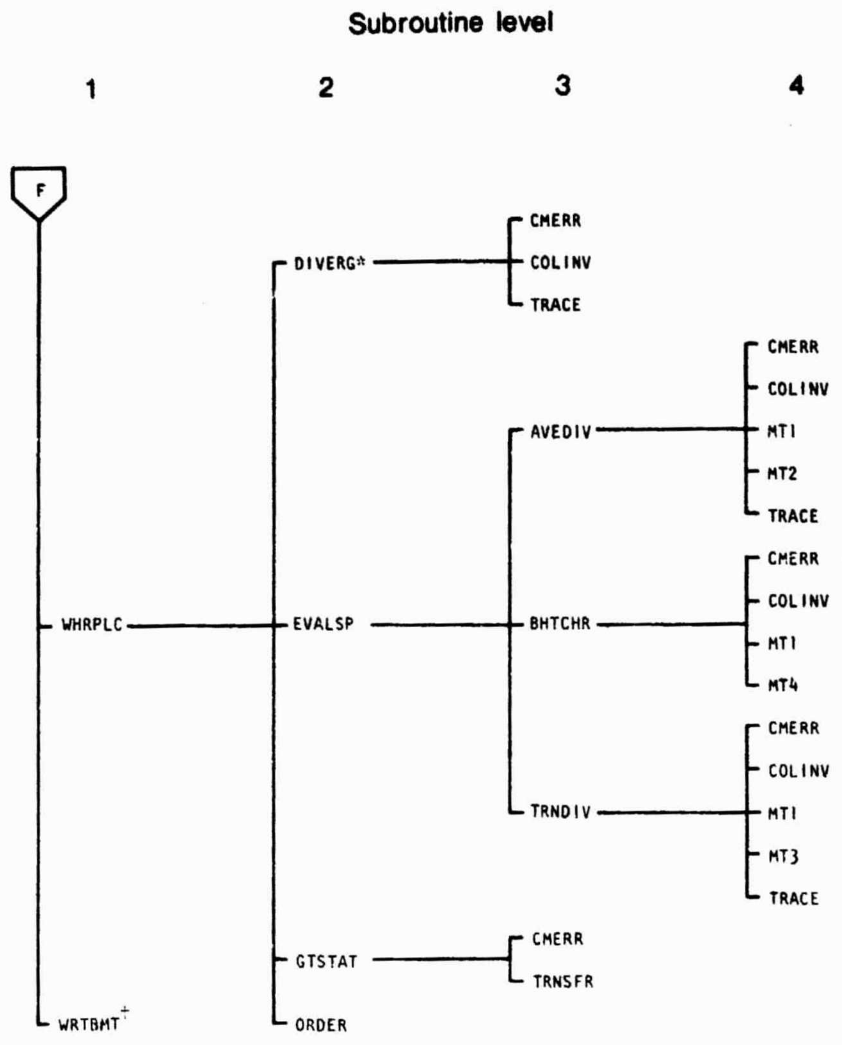


Figure 10-1.- Continued.



*Entry point is DIVRG1.

Figure 10-1.- Continued.



*Entry point is DIVRGI.

†Entry point is WRTBM.

Figure 10-1.- Concluded.

10.1 SELECT

The SELECT subprogram is the driver routine for the SELECT processor.

10.1.1 LINKAGES

The SELECT routine calls the CMERR, DAVIDN, EVLFET, EXSRCH, GENRPT, ORDER, PLOT, PRELIM, SETUP4, USERIN, WHRPLC, and WRTBMT subprograms. It is called by MONTOR.

10.1.2 INTERFACES

The SELECT subprogram interfaces with other routines through common blocks BESTKN, FSL, GLOBAL, and INFORM and through the calling arguments.

10.1.3 INPUTS

Calling sequence: CALL SELECT(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.

10.1.4 OUTPUTS

This subprogram outputs the best separability measure and corresponding features for all channels.

10.1.5 STORAGE REQUIREMENTS

This subprogram requires 55 570 bytes of storage.

10.1.6 DESCRIPTION

The SELECT driver routine coordinates the various subprograms for measuring the relative importance of the individual channels and selecting the set of channels which provides maximum discrimination between subclasses. It calls SETUP4 to read and analyze supervisory control card images and sets the ADRESD parameter for the random-access disk file. Using subprogram PRELIM, it computes separability measures for the full set of user-specified channels using one of the following criteria (specified by a user-input value for the parameter CRIKEY):

- If CRIKEY = 1, weighted interclass divergence
- If CRIKEY = 2, weighted average transformed divergence
- If CRIKEY = 3, weighted average Bhattacharyya distance

According to the value of PRCKEY, one of the following optimization procedures is selected:

- If PRCKEY = 1, the Exhaustive Search procedure (subprogram EXSRCH)
- If PRCKEY = 2, the Without Replacement procedure (subprogram WHRPLC)
- If PRCKEY = 3, the Davidon-Fletcher-Powell procedure (subprogram DAVIDN)
- If PRCKEY = 4, the user-input B-matrix evaluation (subprogram USERIN)
- If PRCKEY = 5, the user-input channel set evaluation (subprogram EVLFET)

If the Davidon-Fletcher-Powell procedure is indicated, either a user-input B-matrix file is read or the Without Replacement procedure is used to find a best set or linear combination of channels.

Subprogram WRTBMT is called to write the B-matrix file (BMFIL), and and subprograms GENRPT and PLOT are called to generate reports and plots.

See volume II, section 10, for a detailed description of the above procedures.

10.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.1.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.2 AVEDIV

The AVEDIV subprogram computes weighted average interclass divergence and partial derivatives with respect to the B-matrix.

10.2.1 LINKAGES

The AVEDIV subprogram calls the CMERR, COLINV, MT1, MT2, and TRACE subprograms. It is called by subprograms EVALSP and PRELIM.

10.2.2 INTERFACES

The AVEDIV subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.2.3 INPUTS

Calling Sequence: CALL AVEDIV(SMSR,COVMTX,S,COVMT2,S2,WRKRY, IWRKSZ,IPART,PARTLS,BMAT,IFULL)

Parameter	Dimension	In/out	Definition
SMSR	1*	Out	Average divergence for all NOFET channels.
COVMTX	VARSZ2,NOCLS2	In	Array containing the subclass covariance matrices; VARSZ2 = size of each covariance matrix; NOCLS2 = number of classes (from SAVTAP) to be used in processing.
S	VARSZ2,NOCLS2	In	Array containing S-matrices used in computing weighted average divergence.
COVMT2	VARSZ4,NOCLS2*	In	Array containing subclass covariance matrices.

*Double precision.

Parameter	Dimension	In/cut	Definition
S2	VARSZ4,NOCLS2*	In	Array containing matrix used in computing weighted average divergence.
WRKRY	1*	Out	Working storage for covariance matrices.
IWRKSZ	1	In	Maximum storage for WRKRY.
IPART	1	In	If = 0, partial derivatives with respect to the B-matrix are computed.
PARTLS	1*	In	Array containing partial derivatives.
BMAT	1*	In	Array containing the B-matrix.
IFULL	1	In	If = 1, average divergence is computed for all NOFET channels and partial derivatives cannot be computed.

10.2.4 OUTPUTS

Not applicable.

10.2.5 STORAGE REQUIREMENTS

This subprogram requires 2716 bytes of storage.

10.2.6 DESCRIPTION

According to the values assigned to the parameters IFULL and IPART, the subprogram AVEDIV computes the average divergence for all channels or the partial derivatives with respect to

*Double precision.

the B-matrix. That is, if IFULL = 1: average divergence for all channels is computed and partial derivatives cannot be computed; if IPART = 0, partial derivatives with respect to the B-matrix are computed. The average interclass divergence is stored in SMSR, and partial derivatives are stored in the array PARTLS.

10.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.2.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.3 BHTCHR

The BHTCHR subprogram computes the interclass Bhattacharyya distance, the weighted average interclass distance, and partial derivatives with respect to the B-matrix.

10.3.1 LINKAGES

The BHTCHR subprogram calls the CMERR, COLINV, MT1, and MT4 subprograms. It is called by the EVALSP and PRELIM subprograms.

10.3.2 INTERFACES

The BHTCHR subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.3.3 INPUTS

Calling sequence: CALL BHTCHR(SMSR,COVMTX,AVEMTX,WEIGHT, DIVTAB,COVMT2,AVEMT2,WRKRY,IWRKSZ,IPART,PARTLS,BMAT,IFULL)

Parameter	Dimension	In/out	Definition
SMSR	1*	Out	Weighted average interclass distance for all NOFET channels.
COVMTX	VARSZ2,NOCLS2	In	Array containing the subclass covariance matrices.
AVEMTX	NOFET2,NOCLS2	In	Array containing subclass mean vectors.
WEIGHT	DIVSIZ	Out	Array containing intersubclass weights.
DIVTAB	DIVSIZ*	Out	Array containing interclass Bhattacharyya distances.

*Double precision.

Parameter	Dimension	In/out	Definition
COVMT2	VARSZ4,NOCLS2*	In	Array containing subclass covariance matrices.
AVEMT2	NOFET4,NOCLS2*	In	Array containing subclass mean vectors.
WRKRY	1*	Out	Working storage for covariance matrices.
IWRKSZ	1	In	Maximum storage for WRKRY.
IPART	1	In	If = 0, partial derivatives with respect to the B-matrix are computed.
PARTLS	1*	Out	Array containing partial derivatives.
BMAT	1*	In	Array containing the B-matrix.
IFULL	1	In	If = 1, the interclass Bhattacharyya distance is computed for all NOFET channels.

10.3.4 OUTPUTS

The results are returned for use by the calling routine.

10.3.5 STORAGE REQUIREMENTS

This subprogram requires 5034 bytes of storage.

10.3.6 DESCRIPTION

According to the values assigned to the parameters IFULL and IPART, the subprogram BHTCHR computes the interclass Bhattacharyya distance and the weighted average distance or partial derivatives

*Double precision.

with respect to the B-matrix. That is, if IFULL = 1: The subprogram computes the inverse and determinant of the covariance matrix for each class I, the inverse and determinant of the covariance matrix for each class J and for the sum of classes I and J, and the interclass and weighted interclass Bhattacharyya distances. If IPART = 0, the subprogram computes partial derivatives; if IPART ≠ 0, partial derivatives are not computed. The subprogram places the results in working storage, computes the weighted average interclass distance, and returns.

10.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.3.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.4 BSTCHK

The BSTCHK subprogram checks the validity of user-requested channels. An additional entry point, EVLCHK, validates user-requested channel numbers and, if necessary, generates a corrected channel request queue.

10.4.1 LINKAGES

The BSTCHK subprogram does not call any other subprogram. It is called by SETUP4.

10.4.2 INTERFACES

The BSTCHK subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.4.3 INPUTS

Calling sequences:

a. CALL BSTCHK(NOBEST)

Parameter	Dimension	In/out	Definition
NOBEST	1	In/out	Number of features to analyze.

b. ENTRY EVLCHK(COMBUF,CPTR)

Parameter	Dimension	In/out	Definition
COMBUF	1	In/out	Array containing list of user-requested channels for comparison with FETVC2 (vector containing correct channels to be used in processing).
CPTR	1	In/out	Number of channels to be evaluated.

10.4.4 OUTPUTS

This subprogram outputs correct features or a correct evaluation request on the specified unit.

10.4.5 STORAGE REQUIREMENTS

This subprogram requires 1534 bytes of storage.

10.4.6 DESCRIPTION

Subprogram BSTCHK checks the vector of user-input channels, generates an error message if input channels exceed the maximum number of channels to be used in processing, and returns a correct set of channels to the calling routine.

Entry EVLCHK validates the user-requested channel numbers by comparing them with the channels stored in FETVC2, generates an error message if the request is invalid, and returns a corrected channel request queue.

10.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.4.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.5 COLINV

The COLINV subprogram inverts a given symmetric positive definite matrix S by computing a triangular factorization R. It inverts R to obtain A and then finds the inverse of S.

10.5.1 LINKAGES

The COLINV subprogram does not call any other subprogram. It is called by AVEDIV, BHTCHR, DIVERG, and TRNDIV within the SELECT processor.

10.5.2 INTERFACES

The COLINV subprogram interfaces with other routines through the calling arguments.

10.5.3 INPUTS

Calling sequence: CALL COLINV(S,N,IERR,IND,DET)

Parameter	Dimension	In/out	Definition
S	1*	In/out	Array containing lower triangular part of the given symmetric matrix stored row-wise in N(N + 1)/2 successive storage locations.
N	1	In	Number of rows or columns in a given matrix.
IERR	1	Out	Resulting error parameter coded as follows: If = 0, no error has occurred; if = -1, matrix S is not positive definite.

*Double precision.

Parameter	Dimension	In/out	Definition
IND	1	In	Parameter indicating which matrix is returned; i.e.: If = 1, matrix R; if = 2, matrix A; and, if = 3, the inverse of the matrix S.
DET	1*	Out	Determinant of returned matrix.

10.5.4 OUTPUTS

Not applicable.

10.5.5 STORAGE REQUIREMENTS

This subprogram requires 1786 bytes of storage.

10.5.6 DESCRIPTION

The COLINV subprogram inverts a given symmetric, positive definite matrix S by computing a triangular factorization R. It inverts R to obtain the lower triangular matrix A and finds the inverse of S. S = the transpose of R; A = the inverse of R; thus, the inverse of S = (the transpose of A) × A.

The program returns the R-matrix, the A-matrix, or the inverse of the S-matrix, according to the value of the input parameter IND. It also returns the determinant of the returned matrix.

10.5.7 FLOW CHART

he available subprogram flow charts for this processor are provided in section 10.36.

10.5.8 LISTING

The subprogram listing is provided in volume IV, section 10.

*Double precision.

10.6 CONVRT

The CONVRT subprogram converts EBCDIC characters to computational characters or computational characters to EBCDIC characters.

10.6.1 LINKAGES

The CONVRT subprogram does not call any other subprogram. It was previously called by PLOT but is now inactive.

10.6.2 INTERFACES

The CONVRT subprogram interfaces with other routines through the calling arguments.

10.6.3 INPUTS

Calling sequence: CALL CONVRT(NOSPAC,IFLG)

Parameter	Dimension	In/out	Definition
NOSPAC	1	In/out	Character to be converted.
IFLG	1	In	Flag indicating if the input character is EBCDIC (≠0) or a computational number (=0).

10.6.4 OUTPUTS

The results are returned for use by the calling routine.

10.6.5 STORAGE REQUIREMENTS

This subprogram requires 792 bytes of storage.

10.6.6 DESCRIPTION

Not required.

10.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.6.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.7 CUBIC

The CUBIC subprogram generates the cubic-fit technique of minimizing a function using the Davidon-Fletcher-Powell method.

10.7.1 LINKAGES

The CUBIC routine does not call any other subprogram. It is called by DAVDN2.

10.7.2 INTERFACES

The CUBIC subprogram interfaces with other routines through the calling arguments.

10.7.3 INPUTS

Calling sequence: CALL CUBIC(XX,YY,XMIN2)

Parameter	Dimension	In/out	Definition
XX	4*	In	Array containing x-coordinates of the function.
YY	4*	In	Array containing y-coordinates of the function.
XMIN2	1*	Out	Minimum value of the cubic fit.

10.7.4 OUTPUTS

The results are returned for use by the calling routine.

10.7.5 STORAGE REQUIREMENTS

This subprogram requires 1808 bytes of storage.

---

*Double precision.

10.7.6 DESCRIPTION

The CUBIC subprogram accepts function values and x-values as input, generates a best polynomial or cubic fit for the function, and finds a minimum value for the cubic fit. The minimum value for the cubic fit, which is an approximation of the minimum value of the function, is returned in the parameter XMIN2.

10.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.7.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.8 DAVDN1

The DAVDN1 subprogram initializes the H- and P-matrices used in the Davidon-Fletcher-Powell procedure.

10.8.1 LINKAGES

The DAVDN1 subprogram calls the CMERR and RWRITE subprograms. It is called by DAVIDN.

10.8.2 INTERFACES

The DAVDN1 subprogram interfaces with other routines through common blocks DVNBLK and FSL and through the calling arguments.

10.8.3 INPUTS

Calling sequence: CALL DAVDN1(FX,P,H)

Parameter	Dimension	In/out	Definition
FX	N*	In	Array containing gradients of the function to be minimized.
P	N*	Out	Array containing the vector corresponding to the search direction (H-matrix times the gradient).
H	N*	Out	Array containing an N-by-N matrix approximating the inverse matrix of partial derivatives.

10.8.4 OUTPUTS

This subprogram places the H and FX arrays in temporary storage.

*Double precision.

10.8.5 STORAGE REQUIREMENTS

This subprogram requires 882 bytes of storage.

10.8.6 DESCRIPTION

The DAVDN1 subprogram initializes the identity matrix H, which approximates the inverse matrix of partial derivatives, and the matrix P, which is obtained by multiplying the gradient FX by the H-matrix. Both H and FX are placed in temporary storage for subsequent use in executing the Davidon-Fletcher-Powell procedure.

DAVDN1 also computes the initial value of the function minus the norm of the gradient with respect to the square of the H-matrix and returns it in DFDK.

10.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.8.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.9 DAVDN2

The DAVDN2 subprogram performs a one-dimensional search for the best k of n channels to use in executing the Davidon-Fletcher-Powell procedure.

10.9.1 LINKAGES

The DAVDN2 subprogram calls the CUBIC and FINT1 subprograms. It is called by DAVIDN.

10.9.2 INTERFACES

The DAVDN2 subprogram interfaces with other routines through common block DVNBLK and through the calling arguments.

10.9.3 INPUTS

Calling sequence: CALL DAVDN2(XBAR,XBARS,P,WRK,IWKSZ,*)

Parameter	Dimension	In/out	Definition
XBAR	N†	In/out	Array containing nominal vector of control parameters at the start of each cycle.
XBARS	N†	Out	Array containing perturbed vector of control parameters in accordance with the Davidon algorithm and search procedures.
P	N†	In	Array containing vector corresponding to the search direction.
WRK	IWKSZ†	In	Array for working storage.
IWKSZ	1	In	Maximum storage for WRK.
*			

†Double precision.

10.9.4 OUTPUTS

Not applicable.

10.9.5

This subprogram requires 5968 bytes of storage.

10.9.6 DESCRIPTION

The DAVDN2 subprogram does a one-dimensional search for the best k of n channels to use in processing. It uses either the cubic-fit or the golden-section technique, or a combination of the two methods, and begins by initializing search variables as follows:

KUBIC – If = 1, the cubic-fit technique is used. If = 0, the golden-section technique is used. If = -1, a technique which combines both methods is used.

EPSCF – Initialized as relative epsilon if the cubic-fit or combination method is used.

EPSGS – Initialized as relative epsilon if the golden-section or combination method is used.

IPART – If = -1, partial derivatives will not be computed.

The program has three parts: Part 1 establishes the golden section in which the function is unimodal. Part 2 reduces the golden section containing the minimum value and checks on the unimodality of the function in the new interval. In part 3, the minimum value of the function is found using the appropriate technique and the y-coordinates are arranged in ascending order. FINT2 is called to evaluate the separability measure and to obtain partial derivatives. Values for each channel are returned in the parameter XBAR.

10.9.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.9.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.10 DAVDN3

The DAVDN3 subprogram updates the H-matrix and the P-matrix for the next cycle through the search for the best k of n channels to be used in executing the Davidon-Fletcher-Powell method.

10.10.1 LINKAGES

The DAVDN3 subprogram calls the RREAD and RWRITE subprograms. It is called by DAVIDN.

10.10.2 INTERFACES

The DAVDN3 subprogram interfaces with other routines through common blocks DVNBLK and FSL and through the calling arguments.

10.10.3 INPUTS

Calling sequence: CALL DAVDN3(FX,FX1,P,H,HY)

Parameter	Dimension	In/out	Definition
FX	N*	In	Array containing the gradient of the function being minimized.
FX1	N*	In	Array containing the value of the gradient at the beginning of the cycle.
P	N*	In	Array containing the vector corresponding to the search direction.
H	N*	In	Array containing the H-matrix approximating the inverse matrix of partial derivatives.
HY	N*	Out	Array containing the vector used in updating the H-matrix.

*Double precision.

10.10.4 OUTPUTS

The results are stored for use by the calling routine.

10.10.5 STORAGE REQUIREMENTS

This subprogram requires 1724 bytes of storage.

10.10.6 DESCRIPTION

The H-matrix is read one row at a time and placed on scratch files for updating and storage. It is used to approximate the inverse matrix of partial derivatives (the P-matrix). The P-matrix is placed in temporary storage for updating and is later stored on scratch files.

10.10.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.10.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.11 DAVIDN

The DAVIDN subprogram is the driver routine for the Davidon-Fletcher-Powell procedure.

10.11.1 LINKAGES

The DAVIDN subprogram calls the BMFIL, CMERR, DAVDN1, DAVDN2, DAVDN3, DIVERG, FINT1, ORDER, RREAD, and RWRITE subprograms. It is called by the SELECT driver routine.

10.11.2 INTERFACES

The DAVIDN subprogram interfaces with other routines through common blocks DVNBLK, FSL, and INFORM and through the calling arguments.

10.11.3 INPUTS

Calling sequence: CALL DAVIDN(COVMTX,AVEMTX,DIVTAB,WEIGHT,COVMT2,AVEMT2,S,S2,BMAT,PARTLS,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	1	In/out	Array containing subclass covariance matrices.
AVEMTX	1	In/out	Array containing subclass mean vectors.
DIVTAB	1*	In/out	Array containing interclass Bhattacharyya distances.
WEIGHT	1	In/out	Array containing intersubclass weights.
COVMT2	1*	In/out	Array containing subclass covariance matrices.

*Double precision.

Parameter	Dimension	In/out	Definition
AVEMT2	1*	In/out	Array containing subclass mean vectors.
S	1	In/out	Array containing S-matrices used in computing weighted average divergence.
S2	1*	In/out	Array containing S-matrices used in computing weighted average divergence.
BMAT	NOFET4,NOFET2*	Out	Array containing the B-matrix.
PARTLS	1*	In/out	Array containing partial derivatives.
WRKRY	1*	In	Working storage array for covariance matrices.
IWRKSZ	1	In	Maximum storage for WRKRY.

10.11.4 OUTPUTS

Not applicable.

10.11.5 STORAGE REQUIREMENTS

This subprogram requires 2900 bytes of storage.

10.11.6 DESCRIPTION

The DAVIDN subprogram utilizes its subprograms to execute the Davidon-Fletcher-Powell procedure for channel selection. See the subprogram flow chart for details.

*Double precision.

10.11.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.11.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.12 DIVERG

The DIVERG subprogram computes average interclass divergence. An additional entry, DIVRG1, provides for the double-precision subclass covariance matrix and subclass mean vector in the computation of interclass divergence.

10.12.1 LINKAGES

The DIVERG subprogram calls the CMERR, COLINV, and TRACE subprograms. It is called by the DAVIDN, EVLFET, EXSRCH, PRELIM, USERIN, and WHRPLC subprograms.

10.12.2 INTERFACES

The DIVERG subprogram interfaces with other routines through the calling arguments.

10.12.3 INPUTS

Calling sequences:

a. CALL DIVERG(COVMTX,VARSIZ,AVEMTX,DIVTAB,NOCLS,NOFET,WRKRY, IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	VARSIZ,NOCLS	In	Array containing subclass covariance matrices.
VARSIZ	1	In	Size of each covariance matrix.
AVEMTX	NOFET,NOCLS	In	Array containing subclass mean vectors.
DIVTAB	1*	Out	Array containing average interclass divergence.
NOCLS	1	In	Number of classes to be used in processing.

*Double precision for DIVERG only.

Parameter	Dimension	In/out	Definition
NOFET	1	In	Number of channels to be used in processing.
WRKRY	1*	In/out	Working storage array for covariance matrices.
IWRKSZ	1	In	Maximum storage for WRKRY.

b. ENTRY DIVRG1(COVMT2,VARSIZ,AVEMT2,DIVTAB,NOCLS,NOFET,WRKRY, IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMT2	VARSIZ,NOCLS†	In	Array containing subclass covariance matrices.
AVEMT2	NOFET,NOCLS†	In	Array containing subclass mean vectors.

10.12.4 OUTPUTS

The results are stored for use by the calling routine.

10.12.5 STORAGE REQUIREMENTS

This subprogram requires 2894 bytes of storage.

10.12.6 DESCRIPTION

The DIVERG subprogram computes the average interclass divergence by finding the inverse of the covariance matrix for each class.

10.12.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

*Double precision for DIVERG only.

†Double precision for DIVRG1 only.

10.12.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.13 EVALSP

The EVALSP subprogram coordinates the routines for computing the separability measure for a particular linear combination or set of features.

10.13.1 LINKAGES

The EVALSP subprogram calls the AVEDIV, BHTCHR, and TRNDIV subprograms. It is called by the EVLFET, EXSRCH, FINT1, USERIN, and WHRPLC subprograms.

10.13.2 INTERFACES

The EVALSP subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.13.3 INPUTS

Calling sequence: CALL EVALSP(SMSR,COVMTX,AVEMTX,S,COVMT2, AVEMT2,S2,DIVTAB,WEIGHT,IPART,PARTLS,BMAT,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
SMSR	1*	In/out	Array containing average divergence for all NOFET channels.
COVMTX	1	In/out	Array containing subclass covariance matrices.
AVEMTX	1	In/out	Array containing the subclass mean vectors.
S	1	In/out	Array containing S-matrices used in computing weighted average divergence.
COVMT2	1*	In/out	Array containing reduced subclass covariance matrices.

*Double precision.

Parameter	Dimension	In/out	Definition
AVEMT2	1*	In/out	Array containing reduced subclass mean vectors.
S2	1*	In/out	Array containing reduced S-matrices.
DIVTAB	1*	In/out	Array containing interclass Bhattacharyya distances.
WEIGHT	1	In/out	Array containing intersubclass weights.
IPART	1	In/out	If ≥ 0, partial derivatives with respect to the B-matrix are computed.
PARTLS	1*	In/out	Array containing partial derivatives.
BMAT	1*	In/out	Array containing the B-matrix.
WRKRY	1	In/out	Working storage for covariance matrices.
IWRKSZ	1	In/out	Maximum storage for WRKRY.

10.13.4 OUTPUTS

Not applicable.

10.13.5 STORAGE REQUIREMENTS

This subprogram requires 1042 bytes of storage.

10.13.6 DESCRIPTION

According to the value of the parameter CRIKEY, the EVALSP subprogram calls one of the following subprograms.

- AVEDIV – to compute the weighted average divergence.
- TRNDIV -- to compute the weighted average transformed divergence.
- BHTCHR – to compute the weighted average Bhattacharyya distance.

*Double precision.

10.13.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.13.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.14 EVLFET

The EVLFET subprogram evaluates user-input channels for selection of the best k of n passes to be used in processing.

10.14.1 LINKAGES

This routine calls the DIVERG, EVALSP, and GTSTAT subprograms. EVLFET is called by the SELECT driver routine.

10.14.2 INTERFACES

The EVLFET subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.14.3 INPUTS

The user inputs a set of channels to be processed.

Calling sequence: CALL EVLFET(COVMTX,AVEMTX,DIVTAB,WEIGHT, COVMT2,AVEMT2,S,S2,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	1	In/out	Array containing subclass covariance matrices.
AVEMTX	1	In/out	Array containing subclass mean vectors.
DIVTAB	1*	In/out	Array containing interclass Bhattacharyya distances.
WEIGHT	1	In/out	Array containing intersubclass weights.
COVMT2	1*	In/out	Array containing subclass covariance matrices.

*Double precision.

Parameter	Dimension	In/out	Definition
AVEMT2	1*	In/out	Array containing subclass mean vectors.
S	1	In/out	Array containing S-matrices used in computing weighted average divergence.
S2	1*	In/out	Array containing S-matrices used in computing weighted average divergence.
WRKRY	1	In/out	Working storage for covariance matrices.
IWRKSZ	1	In/out	Maximum storage for WRKRY.

10.14.4 OUTPUTS

Not applicable.

10.14.5 STORAGE REQUIREMENTS

This subprogram requires 828 bytes of storage.

10.14.6 DESCRIPTION

The EVLFET subprogram calls GTSTAT to select subsets of the covariance matrix and mean vector to use in processing and EVALSP to compute the separability measure for the channel set. If CRIKEY = 1, it calls DIVRG1 to compute the interclass divergence; otherwise, it returns.

10.14.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

*Double precision.

10.14.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.15 EXSRCH

The EXSRCH subprogram uses the Exhaustive Search procedure to find the best k of n channels.

10.15.1 LINKAGES

The EXSRCH subprogram calls the DIVERG, EVALSP, GETSET, and GTSTAT subprograms. It is called by the SELECT driver routine.

10.15.2 INTERFACES

The EXSRCH subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.15.3 INPUTS

Calling sequence: CALL EXSRCH(COVMTX,AVEMTX,DIVTAB,WEIGHT, COVMT2,AVEMT2,S,S2,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOCLS2	In/out	Array containing subclass covariance matrices.
AVEMTX	NOFET2,NOCLS2	In/out	Array containing subclass mean vectors.
DIVTAB	DIVSIZ*	In/out	Array containing interclass Bhattacharyya distance; DIVSIZ = size of interclass separability measure table.
WEIGHT	1	In/out	Array containing intersubclass weights.
COVMT2	VARSZ4,1*	In/out	Array containing subclass covariance matrices.

*Double precision.

Parameter	Dimension	In/out	Definition
AVEMT2	NOFET4,1*	In/out	Array containing subclass mean vectors.
S	VARSZ2,NOCLS2	In/out	Array containing S-matrices used in computing weighted average divergence.
S2	VARSZ4,1*	In/out	Array containing S-matrices used in computing weighted average divergence.
WRKRY	1	In/out	Working storage.
IWRKSZ	1	In/out	Maximum storage for WRKRY.

10.15.4 OUTPUTS

The results are returned for use by the calling routine.

10.15.5 STORAGE REQUIREMENTS

This subprogram requires 1908 bytes of storage.

10.15.6 DESCRIPTION

The EXSRCH subprogram uses the Exhaustive Search procedure to find the best NOFET4 out of NOFET2 channels by maximizing the separability measure indicated by the parameter CRIKEY (see section 10.1.6). It calls GETSET for the set of features, GTSTAT for the required statistics for the set of features, and EVALSP to evaluate the separability measure. Partial derivatives are not calculated if IPART = -1.

10.15.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

*Double precision.

10.15.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.16 FINT1

The FINT1 subprogram is a driver routine for obtaining the partial derivatives and/or the separability measure for a given channel set for use in the Davidon-Fletcher-Powell procedure. The first entry point, FINT1, is used to initialize addresses for the argument list and print the header. Entry FINT2 is called for each evaluation of the separability measure and to obtain partial derivatives.

10.16.1 LINKAGES

The FINT1 subprogram calls the EVALSP and GTSTAT subprograms. It is called by DAVIDN and DAVDN2.

10.16.2 INTERFACES

The FINT1 subprogram interfaces with other routines through common blocks FNTDUM, FSL, GLOBAL, and INFORM and through the calling arguments.

10.16.3 INPUTS

Calling sequences:

a. CALL FINT1(COVMTX,AVEMTX,DIVTAB,WEIGHT,S,S2,COVMT2,AVEMT2, PARTLS)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOCLS2	In	Array containing subclass covariance matrices.
AVEMTX	NOFET2,NOCLS2	In	Array containing subclass mean vectors.
DIVTAB	1*	In/out	Array containing interclass Bhattacharyya distances.

*Double precision.

Parameter	Dimension	In/out	Definition
WEIGHT	1	In/out	Array containing intersubclass weights.
S	VARSZ2,NOCLS2	In/out	Array containing S-matrices used in computing weighted average divergence.
S2	VARSZ4,NOCLS2†	In/out	Array containing S-matrices used in computing weighted average divergence.
COVMT2	VARSZ4,NOCLS2†	In/out	Array containing subclass covariance matrices.
AVEMT2	NOFET4,NOCLS2†	In/out	Array containing subclass mean vectors.
PARTLS	1†	In/out	Array containing partial derivatives.

b. ENTRY FINT2(SPSR,IPART,BMAT,WRKRY,IWRKSZ,*)

Parameter	Dimension	In/out	Definition
SPSR	1	In	Separability measure returned from EVALSP.
IPART	1	In	If $\geq$ 0, partial derivatives with respect to the B-matrix are computed.
BMAT	1†	In/out	Array containing the B-matrix.
WRKRY	1	In/out	Working storage array.
IWRKSZ	1	In/out	Maximum storage for WRKRY.
*	1		Exit route to address indicated by calling program.

†Double precision.

10.16.4 OUTPUTS

This subprogram outputs the convergence characteristics summary for the Davidon-Fletcher-Powell procedure, the requested separability measure, and partial derivatives (if requested).

10.16.5 STORAGE REQUIREMENTS

This subprogram requires 2504 bytes of storage.

10.16.6 DESCRIPTION

Entry FINT2 obtains transformed statistics for the B-matrix by calling GTSTAT and invokes EVALSP to calculate the separability measure and/or obtain partial derivatives. FINT1 is called to output results.

10.16.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.16.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.17 GENRPT

Subprogram GENRPT generates reports on the line printer.

10.17.1 LINKAGES

The GENRPT subprogram calls the RREAD subprogram. It is called by the SELECT driver routine.

10.17.2 INTERFACES

The GENRPT subprogram interfaces with other routines through common blocks FSL, GLOBAL, and INFORM and through the calling arguments.

10.17.3 INPUTS

Calling sequence: CALL GENRPT(CLSNAM,WEIGHT,DIVTAB,WRKRY, IWRKSZ,FETVEC)

Parameter	Dimension	In/out	Definition
CLSNAM	NOCLS2	In	Array containing class names.
WEIGHT	1	In	Array containing intersubclass weights.
DIVTAB	1*	In	Array containing interclass Bhattacharyya distances.
WRKRY	1*	In	Working storage.
IWRKSZ	1	In	Maximum storage for WRKRY.
FETVEC	30	In	Array containing selected features.

*Double precision.

10.17.4 OUTPUTS

The GENRPT subprogram outputs results on the line printer or other specified unit.

10.17.5 STORAGE REQUIREMENTS

This subprogram requires 4614 bytes of storage.

10.17.6 DESCRIPTION

According to the values assigned to the parameters CRIKEY and PRCKEY (section 10.1.6), the GENRPT subprogram generates the printed reports for channel selection activity using the optimization or the separability measure procedure. The printed results include the channels considered, evaluated, and selected; the number of linear combinations; the separability measures for linear combinations, selected channels, and evaluate requests; the minimum and maximum separability measures; the interclass separability table; and the procedure used in channel selection.

10.17.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.17.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.18 GETSET

The GETSET subprogram returns a unique set of indices for the maximum number of features in the VEC array.

10.18.1 LINKAGES

The GETSET subprogram does not call any other subprogram. It is called by EXSRCH.

10.18.2 INTERFACES

The GETSET subprogram interfaces with other routines through the calling arguments.

10.18.3 INPUTS

Calling sequence: CALL GETSET(VEC,NOFET4,NOFET2,LAST)

Parameter	Dimension	In/out	Definition
VEC	1	In/out	Array containing vector of channel numbers.
NOFET4	1	In	Maximum features in VEC.
NOFET2	1	In	Number of channels to be used in processing.
LAST	1	Out	Indicator set = 1 when VEC(NOFET4) $\leq$ NOFET2; otherwise, = 0.

10.18.4 OUTPUTS

The results are returned for use by the calling routine.

10.18.5 STORAGE REQUIREMENTS

This subprogram requires 622 bytes of storage.

10.18.6 DESCRIPTION

The GETSET subprogram receives a set of channels as input and compares and reduces it to a unique set of channels, which is returned in VEC.

The array VEC should be initialized to 1, 2, 3, ···, (NOFET4 - 1) before the routine is entered the first time.

10.18.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.18.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.19 GTSTAT

Subprogram GTSTAT retrieves the reduced feature covariance matrices and mean vectors for each subclass.

10.19.1 LINKAGES

The GTSTAT subprogram calls the CMERR and TRNSFR subprograms. It is called by the EVLFET, EXSRCH, FINT1, USERIN, and WHRPLC subprograms.

10.19.2 INTERFACES

The GTSTAT subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.19.3 INPUTS

Calling sequence: CALL GTSTAT(COVMTX,AVEMTX,S,COVMT2,AVEMT2,S2, VEC,BMAT,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOCLS2	In	Array containing subclass covariance matrices.
AVEMTX	NOFET2,NOCLS2	In	Array containing subclass mean vectors.
S	VARSZ2,NOCLS2	In	Array containing S-matrices used in computing weighted average divergence.
COVMT2	VARSIZ4,1*	Out	Array containing reduced subclass covariance matrices.
AVEMT2	NOFET4,1*	Out	Array containing reduced subclass mean vectors.

*Double precision.

Parameter	Dimension	In/out	Definition
S2	VARSZ4,1*	Out	Array containing reduced S-matrices.
VEC	1	In	Array containing reduced channel set.
BMAT	NOFET4,1*	In	Array containing B-matrix for linear combination of channels (when PRCKEY = 3 or 4).
WRKRY	1*	In/out	Working storage.
IWRKSZ	1	In	Maximum storage in WRKRY.

10.19.4 OUTPUTS

The results are returned for use by the calling routine.

10.19.5 STORAGE REQUIREMENTS

This subprogram requires 1892 bytes of storage.

10.19.6 DESCRIPTION

According to the value of PRCKEY, the GTSTAT subprogram will execute the procedures necessary for channel selection. If the Without Replacement or Exhaustive Search procedure is used, it selects the elements from the reduced channel set in the parameter VEC. If the Davidon-Fletcher-Powell procedure is used, it multiplies the summation of the mean vector times the B-matrix and stores the result in AVEMT2. It calls the TRNSFR subprogram to multiply the covariance matrix times the B-matrix times the transpose of the B-matrix and stores the result in COVMT2. If computation of weighted interclass divergence is requested (CRIKEY = 1), TRNSFR is called again to perform the same calculation on the S-matrix and store the result in S2.

*Double precision.

10.19.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.19.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.20 MT1

The MT1 subprogram multiplies the double-precision matrix A times the single-precision matrix B and stores the result in matrix C in symmetric notation and double precision.

10.20.1 LINKAGES

The MT1 subprogram does not call any other subprogram. It is called by the AVEDIV, BHTCHR, and TRNDIV subprograms.

10.20.2 INTERFACES

The MT1 subprogram interfaces with other routines through the calling arguments.

10.20.3 INPUTS

Calling sequence: CALL MT1(A,B,C,M,N)

Parameter	Dimension	In/out	Definition
A	M,N*	In	Array containing an M-by-N matrix.
B	1	In	Array containing a matrix.
C	M,N*	Out	Working storage.
M	1	In	Number of rows in matrices A and C (NOFET4).
N	1	In	Number of columns in matrices A and C (NOFET2).

10.20.4 OUTPUTS

The results are returned for use by the calling routine.

*Double precision.

10.20.5 STORAGE REQUIREMFNTS

This subprogram requires 764 bytes of storage.

10.20.6 DESCRIPTION

Not required.

10.20.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.20.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.21 MT2

The MT2 subprogram multiplies double-precision matrices A and B and stores the product in matrix C in symmetric notation and double precision.

10.21.1 LINKAGES

The MT2 subprogram does not call any other subprogram. It is called by the AVEDIV subprogram.

10.21.2 INTERFACES

The MT2 subprogram interfaces with other routines through the calling arguments.

10.21.3 INPUTS

Calling sequence: CALL MT2(A,B,C,M,N)

Parameter	Dimension	In/out	Definition
A	1*	In	Array containing a matrix.
B	M,N*	In	Array containing an M-by-N matrix.
C	M,N*	Out	Working storage.
M	1	In	Number of rows in matrices B and C.
N	1	In	Number of columns in matrices B and C.

10.21.4 OUTPUTS

The results are returned for use by the calling routine.

*Double precision.

10.21.5 STORAGE REQUIREMENTS

This subprogram requires 762 bytes of storage.

10.21.6 DESCRIPTION

Not required.

10.21.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.21.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.22 MT3

The MT3 subprogram multiplies and stores matrices A and B in matrix C in symmetric notation and double precision.

10.22.1 LINKAGES

The MT3 subprogram does not call any other subprogram. It is called by the TRNDIV subprogram.

10.22.2 INTERFACES

The MT3 subprogram interfaces with other routines through the calling arguments.

10.22.3 INPUTS

Calling sequence: CALL MT3(A,B,C,L,M,N,ISYMA,ISYMB)

Parameter	Dimension	In/out	Definition
A	1*	In	Array containing a matrix.
B	1*	In	Array containing a matrix.
C	L,N*	Out	Working storage.
L	1	In	If = M, matrix A is stored in symmetric notation.
M	1	In	Number of elements in the matrix to be stored (NOFET4).
N	1	In	If = M, matrix B is stored in symmetric notation.
ISYMA	1	In	If = 1, matrix A is stored in symmetric notation.
ISYMB	1	In	If = 1, matrix B is stored in symmetric notation.

*Double precision.

10.22.4 OUTPUTS

The results are returned for use by the calling routine.

10.22.5 STORAGE REQUIREMENTS

This subprogram requires 1014 bytes of storage.

10.22.6 DESCRIPTION

Two matrices, A and B, are multiplied and the result is stored in matrix C. The input parameters ISYMA and ISYMB indicate whether or not the matrices A and B have been stored in symmetric notation.

10.22.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.22.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.23 MT4

The MT4 subprogram stores a matrix A either in full or symmetric notation and in double precision.

10.23.1 LINKAGES

The MT4 subprogram does not call any other subprogram. It is called by the BHTCHR subprogram.

10.23.2 INTERFACES

The MT4 subprogram interfaces with other routines through the calling arguments.

10.23.3 INPUTS

Calling sequence: CALL MT4(A,B,C,L,M,N,ISYM)

Parameter	Dimension	In/out	Definition
A	1*	In	Array containing a matrix.
B	M,N*	In	Array containing an M-by-N matrix.
C	L,N*	Out	Working storage.
L	1	In	Number of rows in C.
M	1	In	Number of rows in B.
N	1	In	Number of columns in B and C.
ISYM	1	In	If = 1, A is to be stored in symmetric notation; if = 0, A is to be stored in full.

10.23.4 OUTPUTS

The results are returned for use by the calling routine.

10.23.5 STORAGE REQUIREMENTS

This subprogram requires 868 bytes of storage.

10.23.6 DESCRIPTION

Not required.

10.23.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.23.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.24 PLOT

The PLOT subprogram generates a plot of the results of the channel selection process.

10.24.1 LINKAGES

The PLOT subprogram calls the SCALE and SETMRG subprograms. It is called by the SELECT driver routine.

10.24.2 INTERFACES

The PLOT subprogram interfaces with other routines through common blocks FSL and GLOBAL and through the calling arguments.

10.24.3 INPUTS

Calling sequence: CALL PLOT(X,Y,NOX,MAXX,ILABLX,ILABLY,ICODE, IOPT)

Parameter	Dimension	In/out	Definition
X	NOX*	In	Array containing x-coordinates (real numbers).
Y	NOX*	In	Array containing y-coordinates (real numbers).
NOX	1	In	Number of x-coordinates input (≤ 39).
MAXX	1	Out	Maximum value of point on scale to be input (integer).
ILABLX	20	Out	Array containing labels of the x-axis (≤ 78 characters).
ILABLY	20	Out	Array containing labels of the y-axis (≤ 78 characters).

*Double precision.

Parameter	Dimension	In/out	Definition
ICODE	20	Out	Not used on IBM 370/148 system.
IOPT	1	In	Not used on IBM 370/148 system.

10.24.4 OUTPUTS

This subprogram plots and outputs results of channel selection on the line printer.

10.24.5 STORAGE REQUIREMENTS

This subprogram requires 4500 bytes of storage.

10.24.6 DESCRIPTION

Subprogram PLOT begins by setting all x- and y-axis labels to blank. It then finds the maximum value of the point to be input. If weighted average Bhattacharyya distance is requested, it prepares to generate a separability-to-be-gained map; x- and y-axis labels are stored in label array (LABRAY), and variables are set for a printed plot. PLOT calls subprogram SCALE to scale coordinates for the plotting process, locates y- and x-coordinates, builds the 45°-angle line for the plot, and determines if labels have been printed. If labels have been printed, the labels are placed in the FMTARY array for generating each line of the plot. If the Davidon-Fletcher-Powell or user-input B-matrix evaluation procedure was used in channel selection, a linear combination is plotted. If the Exhaustive Search or Without Replacement procedure has been used, a plot of selected channels is written.

10.24.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.24.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.25 PRELIM

Subprogram PRELIM performs some of the preliminary tasks for feature selection. It sets up the separability measure for the full set of channels specified by the user. This measure is either average interclass divergence, average transformed divergence, or average Bhattacharyya distance. It also computes (optionally) the intersubclass weight factors based on the number of pixels per cluster and multiplies the weights by these factors.

10.25.1 LINKAGES

Subprogram PRELIM calls the AVEDIV, BHTCHR, DIVERG, RWRITE, and TRNDIV subprograms. It is called by the SELECT driver routine.

10.25.2 INTERFACES

The PRELIM subprogram interfaces with other routines through common blocks BESTKN, FSL, and INFORM and through the calling arguments.

10.25.3 INPUTS

Calling sequence: CALL PRELIM(COVMTX,AVEMTX,DIVTAB,WEIGHT,S, WRKRY,WRKSIZ)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOCLS2	In/out	Array containing subclass covariance matrices (lower triangular).
AVEMTX	NOFET2,NOCLS2	In/out	Array containing subclass mean vectors.
DIVTAB	DIVSIZ*	In/out	Array containing intersubclass divergences.

*Double precision.

Parameter	Dimension	In/out	Definition
WEIGHT	DIVSIZ	In/out	Array containing intersubclass weights.
S	VARSZ2,NOCLS2	Out	Array containing S-matrices; computed if CRIKEY = 1.
WRKRY	1*	In/out	Working storage.
WRKSIZ	1	In	Computed as 12 000 - SBASE in SELECT; SBASE = 1 if CRIKEY ≠ 1; = 2 + (NOCLS2 × VARSZ2) if CRIKEY = 1.

10.25.4 OUTPUTS

This subprogram stores results on a scratch file (disk) for later printing.

10.25.5 STORAGE REQUIREMENTS

This subprogram requires 6728 bytes of storage.

10.25.6 DESCRIPTION

In performing its preliminary tasks, subprogram PRELIM sets IPART = -1 so that partial derivatives are not computed. According to value of CRIKEY, it calls the following subprograms.

a. If CRIKEY = 1, DIVERG is called to compute average interclass divergence for all features. If DIVERG is called and the parameter SETWGT = 0, PRELIM computes the intersubclass weights and stores them in the parameter WEIGHT. If weights are to be set by default, PRELIM computes the S-matrices to be used in computing weighted average divergence and selects weights for all NOCLS2 classes. Based on CRIKEY = 1, AVEDIV is called, also, to compute weighted average interclass divergence.

*Double precision.

b. If CRIKEY = 2, TRNDIV is called to calculate average weighted transformed divergence for all features.

c. If CRIKEY = 3, BHTCHR is called to calculate the weighted average Bhattacharyya distance.

10.25.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.25.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.26 PRTFLD

The PRTFLD subprogram coordinates the printing of training fields and subclass statistics.

10.26.1 LINKAGES

The PRTFLD subprogram calls the WRTFLD and WRTMTX subprograms. It is called by SETUP4.

10.26.2 INTERFACES

The PRTFLD subprogram interfaces with other routines through common blocks FSL, GLOBAL, and INFORM and through the calling arguments.

10.26.3 INPUTS

Calling sequence: CALL PRTFLD(COVMTX,AVEMTX,FLDMTX,VERTEX,CLSNAM, SUBNAM)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOSUB2	In/out	Array containing subclass covariance matrices.
AVEMTX	NOFET2,NOSUB2	In	Array containing subclass mean vectors.
FLDMTX	4,NOFLD2	In/out	Array containing training field matrix.
VERTEX	2,TOTVT2	In/out	Array containing field vertices.
CLSNAM	NOCLS2	In/out	Array containing class names.
SUBNAM	NOSUB2	In/out	Array containing cluster (subclass) names.

10.26.4 OUTPUTS

This subprogram outputs results on the line printer.

10.26.5 STORAGE REQUIREMENTS

This subprogram requires 1336 bytes of storage.

10.26.6 DESCRIPTION

Subprogram PRTFLD calls the utility subprogram WRTFLD to print training fields. If the parameter STATKY = 0, the program returns control to the calling routine. If STATKY ≠ 0, PRTFLD writes subclass names and means and calls WRTMTX to print the covariance matrix.

10.26.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.26.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.27 SCALE

The SCALE subprogram reduces x- and y-coordinates for the plotting process.

10.27.1 LINKAGES

The SCALE subprogram does not call any other subprogram. It is called by PLOT.

10.27.2 INTERFACES

The SCALE subprogram interfaces with other routines through common block FSL and through the calling arguments.

10.27.3 INPUTS

Calling sequence: CALL SCALE(MAXX,MLNCT,INCRE,YSCLAR,XSCLAR, SCLARY,XLNVLU,YLNVLU,MHORIZ,NOXPT,NOYPT)

Parameter	Dimension	In/out	Definition
MAXX	1	In	Maximum value of point on scale to be input.
MLNCT	1	In	Number of y-coordinates to be scaled.
INCRE	1	In	Increment used in scaling.
YSCLAR	MLNCT	Out	Array containing scaled y-coordinates.
XSCLAR	MHORIZ	Out	Array containing scaled x-coordinates.
SCLARY	8	Out	Array containing scale labels.
XLNVLU	1	Out	Value of each x-coordinate.

Parameter	Dimension	In/out	Definition
YLNVLU	1	Out	Value of each y-coordinate.
MHORIZ	1	In	Number of x-coordinates to be scaled.
NOXPT	1	In	Number of points on the x-axis.
NOYPT	1	In	Number of points on the y-axis.

10.27.4 OUTPUTS

The results are returned for use by the calling routine.

10.27.5 STORAGE REQUIREMENTS

This subprogram requires 1752 bytes of storage.

10.27.6 DESCRIPTION

The SCALE subprogram sets the scale label array SCLARY to zero, determines labels for the x- and y-axes, calculates the value of each point on the x- and y-axes, and returns the results for the plotting process.

10.27.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.27.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.28 SETUP4

Subprogram SETUP4 reads and analyzes control card images and sets options for the SELECT processor.

10.28.1 LINKAGES

The SETUP4 subprogram calls the BMFIL, BSTCHK, CMERR, CRDSTA, FIND12, GRPSCN, NUMBER, NXTCHR, ORDER, PRTFLD, REDSAV, WGTCHK, and WGTSCN subprograms. It is called by the SELECT driver routine.

10.28.2 INTERFACES

The SETUP4 subprogram interfaces with other routines through common blocks BESTKN, FSL, GLOBAL, and INFORM and through the calling arguments.

10.28.3 INPUTS

Input to the SETUP4 subprogram consists of the SAVTAP (or module STAT) file output by the ISOCLS, LABEL, or STAT processor and the BMFIL produced by the SELECT processor.

Calling sequence: CALL SETUP4(ARRAY,TOP,STOPFG,JTIME,SUBRAY, SUBSIZ)

Parameter	Dimension	In/out	Definition
ARRAY	TOP	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
STOPFG	1	Out	Stop flag; if = 1, the $END card has been read.

Parameter	Dimension	In/out	Definition
JTIME	1	In	Counter for the number of times SETUP4 is called from SELECT. If = 1, the line printer output is labeled with the HEADER array contents; if ≠ 1, the heading is not placed on the output.
SUBRAY	1	Out	Utility storage vector.
SUBSIZ	1	In	Maximum usable storage in SUBRAY; set = 12 000 in SELECT.

The control cards relevant to this routine are given in section 10 (table 10-1) of volume II of this user guide.

10.28.4 OUTPUTS

This subprogram outputs results to the SAVTAP file and on the printer and stores results in the common blocks.

10.28.5 STORAGE REQUIREMENTS

This subprogram requires 9306 bytes of storage.

10.28.6 DESCRIPTION

Subprogram SETUP4 reads and analyzes control cards and reports errors and supervises the SELECT processor as to options, criteria, and procedures to be used in selecting the best set of channels. If the B-matrix file is input, it is read and stored using subprogram BMFIL. The module STAT file is read and stored on the SAVTAP. SETUP4 retrieves, reduces, and prints statistics (using subprogram REDSAV); prints user requests and training fields; sets up an array for storage of interclass/subclass weights, subclass descriptions, covariance matrices, mean vectors, and the interclass/subclass separability measure table. Default may be taken on calculation of interclass/subclass weights, in which event weights are computed in PRELIM.

10.28.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.28.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.29 TRACE

The TRACE function computes the trace of the product of two symmetric matrices and stores the result in symmetric notation and double precision.

10.29.1 LINKAGES

The TRACE function does not call any other subprogram. It is called by the AVEDIV, DIVERG, and TRNDIV subprograms.

10.29.2 INTERFACES

The TRACE function interfaces with other routines through the calling arguments.

10.29.3 INPUTS

Calling sequence: TRACE(A,B,N)

Parameter	Dimension	In/out	Definition
A	1*	In	Array containing an N-by-(N+1)/2 matrix.
B	1*	In	Array containing an N-by-(N+1)/2 matrix.
N	1*	In	Number of rows in matrices A and B.

10.29.4 OUTPUTS

Not applicable.

10.29.5 STORAGE REQUIREMENTS

This subprogram requires 620 bytes of storage.

*Double precision.

10.29.6 DESCRIPTION

Function TRACE multiplies matrix A by matrix B and returns the product in symmetric notation in TRACE.

10.29.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.29.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.30 TRNDIV

The TRNDIV subprogram computes the average weighted transformed divergence and partial derivatives with respect to the B-matrix.

10.30.1 LINKAGES

The TRNDIV subprogram calls the CMERR, COLINV, MT1, MT3, and TRACE subprograms. It is called by EVALSP and PRELIM.

10.30.2 INTERFACES

The TRNDIV subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.30.3 INPUTS

Calling sequence: CALL TRNDIV(SPMSR,COVMTX,AVEMTX,COVMT2,AVEMT2, WEIGHT,DIVTAB,WRKRY,IWRKSZ,IPART,PARTLS,BMAT,IFULL)

Parameter	Dimension	In/out	Definition
SPMSR	1	Out	Array containing the average weighted transformed inter-subclass divergence.
COVMTX	VARSZ2,NOCLS2	In	Array containing subclass covariance matrices.
AVEMTX	NOFET2,NOCLS2	In	Array containing subclass mean vectors.
COVMT2	VARSZ4,NOCLS2*	In	Array containing reduced sub-class covariance matrices.
AVEMT2	NOFET4,NOCLS2*	In	Array containing reduced sub-class mean vectors.
WEIGHT	DIVSIZ	In	Array containing intersubclass weights.

*Double precision.

Parameter	Dimension	In/out	Definition
DIVTAB	DIVSIZ*	Out	Array containing intersubclass divergence.
WRKRY	1*	Out	Working storage.
IWRKSZ	1	In	Maximum usable storage in WRKRY.
IPART	1	In	If = 0, partial derivatives with respect to the B-matrix are computed.
PARTLS	1*	Out	Array containing partial derivatives.
BMAT	1*	In/out	Array containing the B-matrix.
IFULL	1	In	If = 1, average divergence is computed for all NOFET channels and partial derivatives cannot be computed.

10.30.4 OUTPUTS

The results are returned for use by the calling routine.

10.30.5 STORAGE REQUIREMENTS

This subprogram requires 5266 bytes of storage.

10.30.6 DESCRIPTION

Subprogram TRNDIV computes either the average weighted transformed divergence (if IFULL = 1) or partial derivatives with respect to the B-matrix (if IFULL $\neq$ 1 and IPART $\geq$ 0).

For the computation of average weighted transformed divergence, TRNDIV begins by initializing variables. If partial derivatives are to be computed, it zeros the array PARTLS. For both processes,

*Double precision.

it tests the size of working storage (IWRKSZ) and generates an error message if storage is insufficient. Calculations are begun by placing the covariance matrix for class I in working storage and calling COLINV to find its inverse; an error message is generated if storage is insufficient. If IFULL = 1, it places the mean vector for all NOFET channels in T and the sum of the covariance matrix and mean vector in working storage. The weighted transformed intersubclass divergence is computed, divided by the number of classes, and the result placed in SPMSR.

If IFULL ≠ 1 and IPART ≥ 0, the full S-matrix for classes I and J and the inverse of the matrix for class J are calculated, and partial derivatives are computed using subprograms MT1 and MT3. The results are placed in PARTLS.

10.30.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.30.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.31 TRNSFR

The TRNSFR subprogram multiplies the B-matrix times its transpose times a matrix A and stores the result in double precision.

10.31.1 LINKAGES

The TRNSFR subprogram does not call any other subprogram. It is called by GTSTAT.

10.31.2 INTERFACES

The TRNSFR subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.31.3 INPUTS

Calling sequence: CALL TRNSFR(A,A2,W,BMAT)

Parameter	Dimension	In/out	Definition
A	VARSZ2,NOCLS2	In	Array containing a matrix.
A2	VARSZ4,NOCLS2*	Out	Storage array for the product of A × BMAT × BMAT^T.
W	NOFET4,NOFET2*	Out	Array for storage of product of BMAT × A.
BMAT	NOFET4,NOFET2*	In	Array containing the B-matrix.

10.31.4 OUTPUTS

The results are returned for use by the calling routine.

10.31.5 STORAGE REQUIREMENTS

This subprogram requires 1236 bytes of storage.

*Double precision.

10.31.6 DESCRIPTION

Not required.

10.31.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.31.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.32 USERIN

The USERIN subprogram coordinates the routines required to compute the separability measure for the user-input B-matrix.

10.32.1 LINKAGES

The USERIN subprogram calls the BMFIL, DIVERG, EVALSP, and GTSTAT subprograms. It is called by the SELECT driver routine.

10.32.2 INTERFACES

The USERIN subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.32.3 INPUTS

Calling sequence: CALL USERIN(COVMTX,AVEMTX,DIVTAB,WEIGHT, COVMT2,AVEMT2,S,S2,BMAT,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOCLS2	In/out	Array containing subclass covariance matrices.
AVEMTX	NOFET2,NOCLS2	In/out	Array containing subclass mean vectors.
DIVTAB	1*	In/out	Array containing intersubclass divergence.
WEIGHT	1	In/out	Array containing intersubclass weights.
COVMT2	VARSZ4,1*	In/out	Array containing reduced subclass covariance matrices.
AVEMT2	NOFET4,1*	In/out	Array containing reduced subclass mean vectors.

*Double precision.

Parameter	Dimension	In/out	Definition
S	VARSZ2,NOCLS2	In/out	Array containing S-matrices for computing weighted average divergence.
S2	VARSZ4,1*	In/out	Array containing reduced S-matrices.
BMAT	1*	Out	Array containing the B-matrix.
WRKRY	1	In/out	Working storage.
IWRKSZ	1	In/out	Maximum storage in WRKRY.

10.32.4 OUTPUTS

Not applicable.

10.32.5 STORAGE REQUIREMENTS

This subprogram requires 1252 bytes of storage.

10.32.6 DESCRIPTION

Subprogram USERIN calls BMFIL to retrieve the B-matrix in single precision and stores it in double precision; calls GTSTAT to get transformed covariance matrices and mean vectors; calls EVALSP to evaluate the separability measure; and, optionally (if CRIKEY = 1), calls DIVRG1 to evaluate interclass divergence.

10.32.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.32.8 LISTING

The subprogram listing is provided in volume IV, section 10.

*Double precision.

10.33 WGTCHK

The WGTCHK subprogram sets up a table of intersubclass weights for the SETUP4 subprogram.

10.33.1 LINKAGES

Subprogram WGTCHK does not call any other subprogram. It is called by the SETUP4 subprogram.

10.33.2 INTERFACES

The WGTCHK subprogram interfaces with other routines through the calling arguments.

10.33.3 INPUTS

Calling sequence: CALL WGTCHK(WEIGHT,CLSNAM NAMPR,WGHT,WPTR, WRKRY,NOCLS2)

Parameter	Dimension	In/out	Definition
WEIGHT	1	In/out	On input, WEIGHT(1) is used to convey the SETWGT flag from SETUP4 to WGTCHK. WEIGHT(2) is used to convey the WTKEY from SETUP4 to WGTCHK. The table of weights for subclass pairs is output in this argument.
CLSNAM	NOCLS2	In	Final set of subclass names from the SAVTAP file.
NAMPR	2,WPTR	In	Subclass names or name pairs input to SETUP4 on WEIGHTS control card images. NAMPR(1,WPTR) = subname$_1$ and NAMPR(2,WPTR) = subname$_2$, if input; otherwise, = blank.

Parameter	Dimension	In/out	Definition
			NAMPR(1,WPTR) may also be input via WEIGHTS OTHERS card image.
WGHT	1	In	Subclass, subclass pair, or OTHERS weights from WEIGHTS card images are input to SETUP4 via the WGTSCN subprogram.
WPTR	1	In	The number of user-input weights.
WRKRY	NOCLS2,NOCLS2	In	Working storage.
NOCLS2	1	In	Total number of subclasses for classes.

The control cards relevant to this routine are described in volume II, section 10 (table 10-1).

10.33.4 OUTPUTS

The results are returned for use by the calling routine.

10.33.5 STORAGE REQUIREMENTS

This subprogram requires 1982 bytes of storage.

10.33.6 DESCRIPTION

According to the values of WTKEY and SETWGT, which are input via calling arguments WEIGHT(1) and WEIGHT(2), subprogram WGTCHK performs one of two functions.

a. If WTKEY $\neq$ 1, it initializes all intersubclass pair weights either = 1.0, or to the value on the WEIGHTS OTHERS card, if input to SETUP4, and places the values in WRKRY. Then, if SETWGT = 2, the subprogram overrides the preset weights in WRKRY with weights from the input WGHT array, using subclass names in the input NAMPR array to direct the storage of input weights. When WRKRY is completed, the subclass-pair weight values are transferred to the weight table (WEIGHT) and returned via calling argument.

b. If WTKEY = 1, WGTCHK assumes that WRKRY contains subclass-pair weights as initialized in internal subroutine INTWGT (SETUP4). Initialization of WRKRY is bypassed, and the flag SETWGT is tested. If SETWGT = 2, the weights in WRKRY are overridden by weights from the input WGHT array, using subclass names in the input NAMPR array to direct the storage of weights into WRKRY. If SETWGT ≠ 2, the weights in WRKRY remain as initialized in INTWGT. The weights from WRKRY are then transferred to the weight table (WEIGHT) and are returned via calling argument.

In the verification process, if a subclass name from the SAVTAP file does not match up with the appropriate subclass name in array NAMPR, an error message is generated and the associated weight is ignored.

10.33.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.33.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.34 WGTSCN

The WGTSCN subprogram scans the WEIGHTS control card and saves class name pairs and associated weights for later verification in subroutine WGTCHK.

10.34.1 LINKAGES

The WGTSCN subprogram calls the FIND12, FLTNUM, and NXTCHR subprograms. It is called by SETUP4.

10.34.2 INTERFACES

The WGTSCN subprogram interfaces with other routines through the calling arguments.

10.34.3 INPUTS

Calling sequence: CALL WGTSCN(CARD,COL,NAMPR,WGHT,WSIZ,NCNT)

Parameter	Dimension	In/out	Definition
CARD	1	In	Array of characters to be scanned.
COL	1	In/out	Column in CARD to begin scan. As output, it is the last column of card scanned.
NAMPR	2,1	Out	Array containing pairs of class names scanned from CARD.
WGHT	1	In/out	Array containing weight for corresponding class pair.
WSIZ	1	In	Size of WGHT buffer.
NCNT	1	In/out	Number of pairs scanned.

The control cards relevant to this routine are given in volume II, section 10 (table 10-1).

10.34.4 OUTPUTS

The results are returned for use by the calling routine.

10.34.5 STORAGE REQUIREMENTS

This subprogram requires 1754 bytes of storage.

10.34.6 DESCRIPTION

Subprogram WGTSCN scans the control card images for class weights, using utility functions NXTCHR, FIND12, and FLTNUM. The WEIGHTS card may be in one of the following forms:

```
WEIGHT      CLASS1=10.5,CLASS2=12.0,OTHERS=20.0

WEIGHTS     (CLASS1,CLASS2)=15.0,CLASS3=1.0,OTHERS=5.0
```

The subprogram reads four characters, then looks for an = or a comma. If an = is found, it looks for a real number; if a comma, it searches for another name.

10.34.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 10.36.

10.34.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.35 WHRPLC

The WHRPLC subprogram coordinates the routines for finding the best k of n features using the Without Replacement procedure.

10.35.1 LINKAGES

The WHRPLC subprogram calls the DIVERG, EVALSP, GTSTAT, and ORDER subprograms. It is called by the SELECT driver routine.

10.35.2 INTERFACES

The WHRPLC subprogram interfaces with other routines through common blocks FSL and INFORM and through the calling arguments.

10.35.3 INPUTS

Calling sequence: CALL WHRPLC(COVMTX,AVEMTX,DIVTAB,WEIGHT, COVMT2,AVEMT2,S,S2,WRKRY,IWRKSZ)

Parameter	Dimension	In/out	Definition
COVMTX	1	In/out	Array containing subclass covariance matrices.
AVEMTX	1	In/out	Array containing subclass mean vectors.
DIVTAB	1*	In/out	Array containing interclass Bhattacharyya distances.
WEIGHT	1	In/out	Array containing intersubclass weights.
COVMT2	1*	In/out	Array containing reduced subclass covariance matrices.
AVEMT2	1*	In/out	Array containing reduced subclass mean vectors.

*Double precision.

Parameter	Dimension	In/out	Definition
S	1	In/out	Array containing S-matrices used in computing weighted average divergence.
S2	1*	In/out	Array containing reduced S-matrices.
WRKRY	1	In/out	Working storage.
IWRKSZ	1	In/out	Maximum storage for WRKRY.

10.35.4 OUTPUTS

The results are returned for use by the calling routine.

10.35.5 STORAGE REQUIREMENTS

This subprogram requires 2434 bytes of storage.

10.35.6 DESCRIPTION

Subprogram WHRPLC determines the best k of n features in the following manner. It saves the value of NOFET4 and selects the channel which extremizes the separability measure. The remaining channels are placed in a vector and compared with the best feature (NBEST). Each feature is compared with the "best so far," and the one which gives the maximum separability measure is saved. WHRPLC than computes interclass measures for the features chosen (using subprograms GTSTAT and EVALSP) and, optionally (if CRIKEY = 1), weighted interclass divergence (using DIVRG1).

10.35.7 FLOW CHART

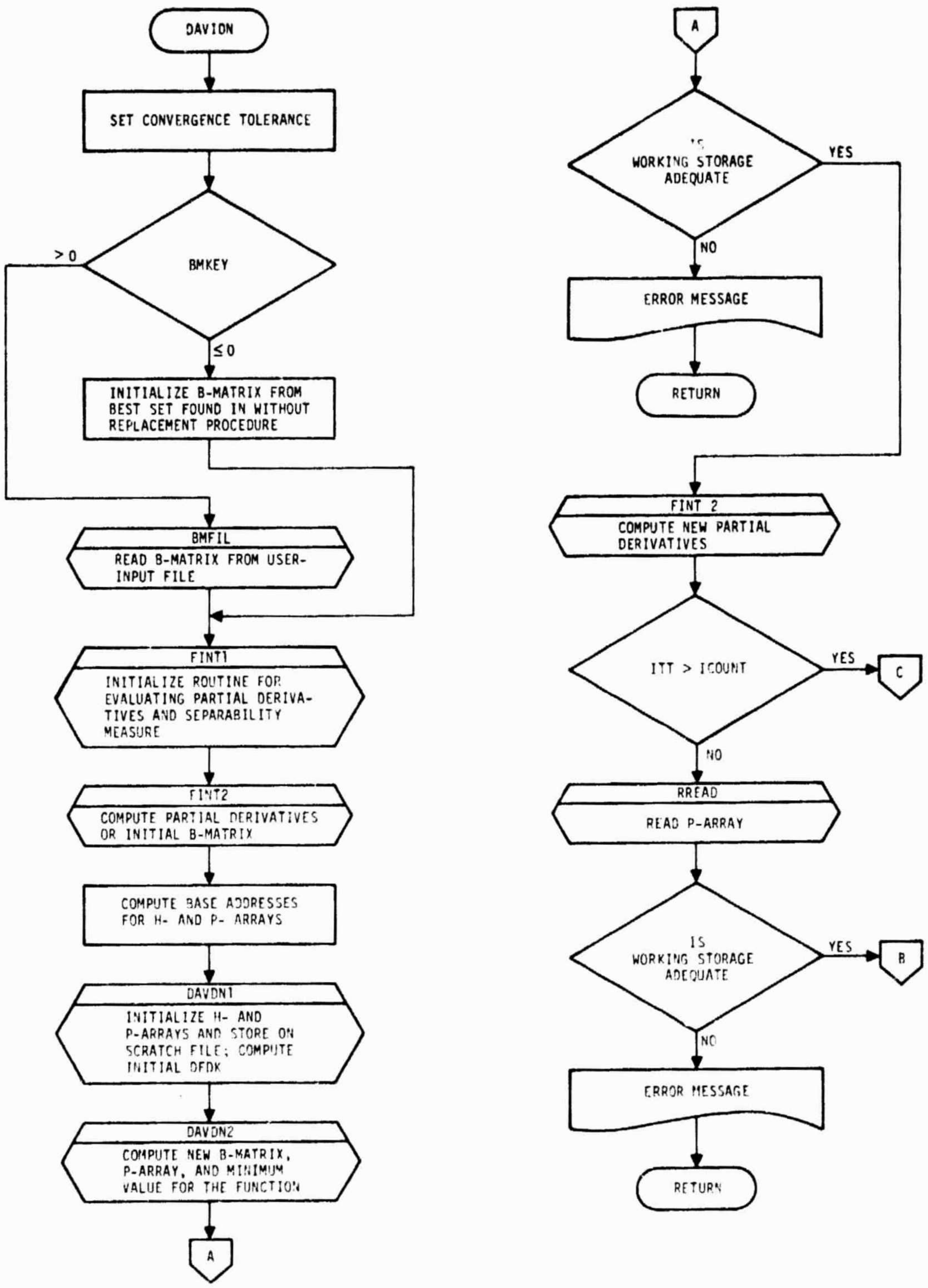
The available subprogram flow charts for this processor are provided in section 10.36.

*Double precision.

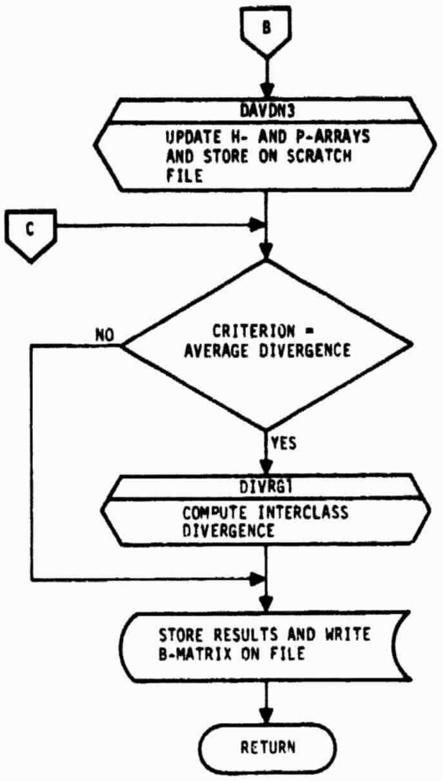
10.35.8 LISTING

The subprogram listing is provided in volume IV, section 10.

10.36 SUBPROGRAM FLOW CHARTS



DAVION
SET CONVERGENCE TOLERANCE
BMKEY
> 0
≤ 0
INITIALIZE B-MATRIX FROM BEST SET FOUND IN WITHOUT REPLACEMENT PROCEDURE
BMFIL
READ B-MATRIX FROM USER-INPUT FILE
FINT1
INITIALIZE ROUTINE FOR EVALUATING PARTIAL DERIVATIVES AND SEPARABILITY MEASURE
FINT2
COMPUTE PARTIAL DERIVATIVES OR INITIAL B-MATRIX
COMPUTE BASE ADDRESSES FOR H- AND P- ARRAYS
DAVDN1
INITIALIZE H- AND P-ARRAYS AND STORE ON SCRATCH FILE; COMPUTE INITIAL DFDK
DAVDN2
COMPUTE NEW B-MATRIX, P-ARRAY, AND MINIMUM VALUE FOR THE FUNCTION
A
A
IS WORKING STORAGE ADEQUATE
YES
NO
ERROR MESSAGE
RETURN
FINT 2
COMPUTE NEW PARTIAL DERIVATIVES
ITT > ICOUNT
YES
C
NO
RREAD
READ P-ARRAY
IS WORKING STORAGE ADEQUATE
YES
B
NO
ERROR MESSAGE
RETURN


B
DAVDN3
UPDATE H- AND P-ARRAYS AND STORE ON SCRATCH FILE
C
CRITERION = AVERAGE DIVERGENCE
NO
YES
DIVRG1
COMPUTE INTERCLASS DIVERGENCE
STORE RESULTS AND WRITE B-MATRIX ON FILE
RETURN

11. CLASSIFY PROCESSOR SUBPROGRAMS

The CLASSIFY processor classifies MSS image data based on statistics (mean vectors and covariance matrices) computed from training fields. Using the statistics for the subclass of interest, the processor assigns each data point within the defined classification field to a subclass. To accomplish this, it utilizes one of the following procedures.

- Standard m-class maximum likelihood classification rule (subprogram CLSFY1).
- Category definition using the sum-of-normal-densities classification rule (subprogram CLSFY2).

The CLASSIFY processor has 16 subprograms that are exclusive to the processor and, in addition, calls 27 utility subprograms to assist in processing. Figure 11-1 is a linkage diagram of the CLASSIFY processor.

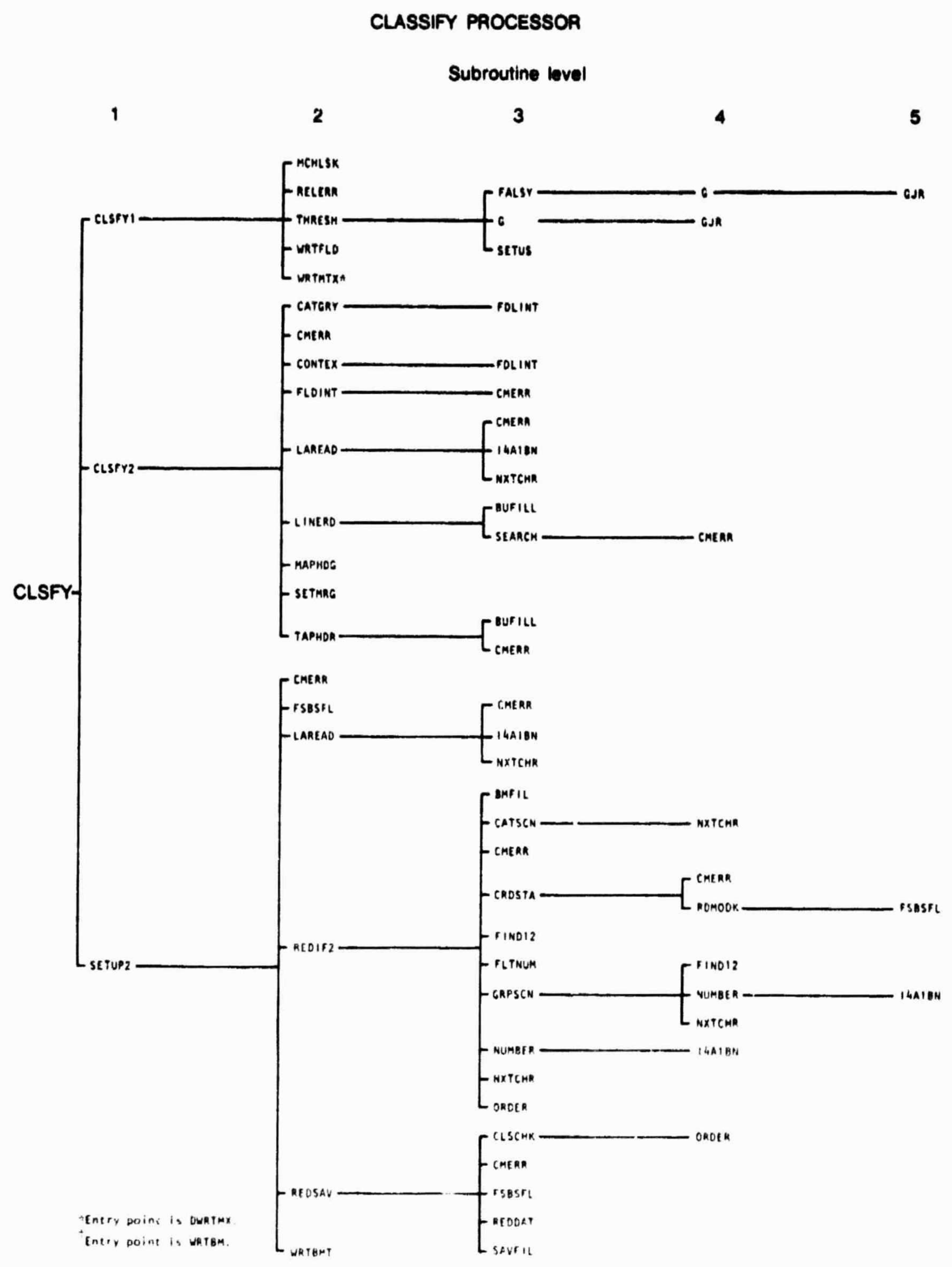


Figure 11-1.- Linkage diagram for the CLASSIFY processor.

11.1 CLSFY

The CLSFY subprogram is the driver routine for the CLASSIFY processor.

11.1.1 LINKAGES

The CLSFY subprogram calls the CLSFY1, CLSFY2, and SETUP2 subprograms. It is called by MONTOR.

11.1.2 INTERFACES

The CLSFY subprogram interfaces with other routines through common blocks BMTRX, CLASS, GLOBAL, INFORM, and SCRACH and through the calling arguments.

11.1.3 INPUTS

Calling sequence: CALL CLSFY(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	3000	In/out	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In/out	Maximum usable storage in ARRAY.

11.1.4 OUTPUTS

Not applicable.

11.1.5 STORAGE REQUIREMENTS

This subprogram requires 1396 bytes of storage.

11.1.6 DESCRIPTION

The CLSFY routine coordinates the various subprograms for classification. It calls SETUP2 for the analysis of supervisory information and the reduction of statistics for processing. CLSFY1 and CLSFY2 are called to perform the classification and output results.

11.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.1.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.2 CATGRY

The CATGRY subprogram performs category classification. It assigns each pixel first to a class and then to a subclass and outputs the chosen subclass number and its corresponding PDF on the MAPTAP file.

11.2.1 LINKAGES

The subroutine CATGRY calls the FDLINT subprogram. It is called CLSFY2.

11.2.2 INTERFACES

The CATGRY subprogram interfaces with other routines through common blocks CLASS and INFORM and through the calling arguments.

11.2.3 INPUTS

Calling sequence: CALL CATGRY(NCHAN,NPTS,AVE,COR,IR,VR,BMATR, IDATA,NLINE,VERTCS,NV,PTSTHS)

Parameter	Dimension	In/out	Definition
NCHAN	1	In	Number of channels.
NPTS	1	In	Number of points in the rectangular field.
AVE	1	In	Array containing means.
COR	1	In	Array containing covariances.
IR	1000	In	Array containing classified data (subclass numbers to which pixels were assigned).
VR	1000	In	Corresponding PDF's of the classified array (IR).

Parameter	Dimension	In/out	Definition
BMATR	BMCOMB,BMFEAT	In	B-matrix, if available. (BMCOMB = number of linear combinations, and BMFEAT = number of channels used in the B-matrix.)
IDATA	1	In	Scan line of data to be classified.
NLINE	1	In/out	Line number corresponding to data tape.
VERTCS	1	In/out	Vertices of field to be classified.
NV	1	In/out	Number of vertices.
PTSTHS	1	In/out	Number of points thresholded.

11.2.4 OUTPUTS

This subprogram outputs results to the MAPTAP file.

11.2.5 STORAGE REQUIREMENTS

This subprogram requires 3580 bytes of storage.

11.2.6 DESCRIPTION

The CATGRY subprogram assigns each pixel to a category and then to a subclass within the category. The subclass number and corresponding PDF are output to the MAPTAP file.

CATGRY performs the following functions:

a. Zeros out IR and VR for storage of new data values.

b. Calls FDLINT to obtain ordered pixel intercepts.

c. Applies the B-matrix, if applicable.

d. Finds the maximum value of the PDF over all subclasses within a category and saves the subclass number of the largest PDF.

e. Finds the maximum category PDF (sum of all PDF's of the chosen subclass within the category).

f. Thresholds subclass PDF's that are too small.

g. Stores the largest subclass PDF and the subclass number of the category with the maximum PDF.

11.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.2.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.3 CATSCN

The CATSCN subprogram is a function which reads the category card from CLSFY and stores category and class names.

11.3.1 LINKAGES

The CATSCN subprogram calls the NXTCHR subprogram. It is called by REDIF2.

11.3.2 INTERFACES

The CATSCN subprogram interfaces with other routines through the calling arguments.

11.3.3 INPUTS

Calling sequence: CATSCN(CARD,KCLSNA,CATNME,KK,NOCLSS,NOCAT)

Parameter	Dimension	In/out	Definition
CARD	62	In	Category control card.
KCLSNA	1	Out	Class names in the order extracted from CARD.
CATNME	1	Out	Category names.
KK	1	In/out	Number of cards to scan.
NOCLSS	1	In/out	Number of classes from the SAVTAP file to be used for processing.
NOCAT	1	-	Number of categories to be used for processing (not used).

11.3.4 OUTPUTS

The results are returned for use by the calling routine.

11.3.5 STORAGE REQUIREMENTS

This subprogram requires 1306 bytes of storage.

11.3.6 DESCRIPTION

CATSCN reads one character at a time from the category card image. When a complete word (six characters) is read, it determines if the word contains a category or class name; if a category name, it is stored in CATNAM; and, if a class name, it is stored in KCLSNA. The subprogram returns when all card images have been read.

11.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.3.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.4 CLSFY1

The CLSFY1 subprogram performs subclass classification.

11.4.1 LINKAGES

The CLSFY1 subroutine calls the MCHLSK, RELERR, THRESH, WRTFLD, and WRTMTX subprograms. It is called by the CLSFY driver routine.

11.4.2 INTERFACES

The CLSFY1 subprogram interfaces with other routines through common blocks CLASS, GLOBAL, INFORM, and SCRACH and through the calling arguments.

11.4.3 INPUTS

Calling sequence: CALL CLSFY1(COVMTX,AVEMTX,FLDMTX,CLSMTX, APRIOR,BMATR,VERTEX,SUBDES,SUBNO,COVNEW,AVENEW,KATNO)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOSUB2	In	Array containing covariance matrices (symmetric storage) for NOCLS2 training classes. $\left[\text{VARSZ2} = \text{NOFET2}\left(\frac{\text{NOFET2} + 1}{2}\right),\right.$ NOFET2 = number of channels, and NOSUB2 = number of subclasses from the SAVTAP file to be used in processing.]
AVEMTX	NOFET2,NOSUB2	In	Array containing NOCLS2 training class mean vectors (NOFET2 means per class).
FLDMTX	4,NOFLD2	In/out	Array of training field matrices. (NOFLD2 = total number of training fields.)

Parameter	Dimension	In/out	Definition
CLSMTX	1	In	Matrix of training class names.
APRIOR	1	In	Array of a priori probability values for each class.
BMATR	BMCOMB,BMFEAT	In	B-matrix, if available.
VERTEX	2,TOTVT2	In/out	Array containing vertices of saved training fields.
SUBDES	1	In	Array of subclass names.
SUBNO	1	In	Array containing number of subclasses in each class.
COVNEW	BMFLG,NOSUB2	Out	Array for storage of B-transformed covariance matrices. (BMFLG = indicator that B-matrix has been input.)
AVENEW	BMCOMB,BMFEAT	Out	Array for storage of B-transformed means.
KATNO	1	In	Category-class correspondence array (category to which each class was assigned).

11.4.4 OUTPUTS

This subprogram outputs results to the MAPTAP file.

11.4.5 STORAGE REQUIREMENTS

This subprogram requires 6182 bytes of storage.

11.4.6 DESCRIPTION

The CLSFY1 subprogram performs the following steps in the classification process.

a. Processes the statistical data (mean vector and covariance matrix) and writes out training field information, classes, and channels (with spectral bands) to be considered in classification.

b. If the B-matrix is available, transforms the covariance matrix and mean vector.

c. Obtains the subclass-pair thresholds (using subprogram THRESH).

d. Outputs the original mean vector and covariance matrix for all subclasses (after transformation of the B-matrix, if available).

e. Obtains the modified Cholesky decomposition and factorization of the covariance matrix for each subclass (using subprogram MCHLSK).

f. Outputs results of the modified Cholesky factorization.

11.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.4.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.5 CLSFY2

The CLSFY2 subprogram performs classification by grouping subclasses into fields or categories.

11.5.1 LINKAGES

The CLSFY2 subprogram calls the CATGRY, CMERR, CONTEX, FLDINT, LAREAD, LINERD, MAPHDG, SETMRG, and TAPHDR subprograms. It is called by the CLSFY driver routine.

11.5.2 INTERFACES

The CLSFY2 subprogram interfaces with other routines through common blocks CLASS, GLOBAL, INFORM, and SCRACH and through the calling arguments.

11.5.3 INPUTS

Calling sequence: CALL CLSFY2(COVMTX,AVEMTX,FLDMTX,CLSMTX, SUBDES,SUBNO,KATNO,BMATRX)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2,NOSUB2	In/out	Array containing covariance matrices (symmetric storage) for NOCLS2 training classes.
AVEMTX	NOFET2,NOSUB2	In/out	Array containing training class mean vector (NOFET2 means per class).
FLDMTX	4,NOFLD2	In	Array containing training field matrices.
CLSMTX	1	In/out	Matrix of training class names.
SUBDES	1	In/out	Array of subclass names.
SUBNO	1	In/out	Array containing number of subclasses in each class.

Parameter	Dimension	In/out	Definition
KATNO	1	In/out	Category-class correspondence array (category to which each class was assigned).
BMATRX	BMCOMB,BMFEAT	In/out	Location of the B-matrix, if available, for transforming input sample vector in subroutine CONTEX.

11.5.4 OUTPUTS

This subprogram outputs classification results to the MAPTAP file.

11.5.5 STORAGE REQUIREMENTS

This subprogram requires 12 734 bytes of storage.

11.5.6 DESCRIPTION

The CLSFY2 subprogram reads the field definition card images containing line-sample coordinates of the fields to be classified and classifies fields with assistance from subroutine CONTEX (the standard classifier) and subroutine CATGRY (the category classifier).

11.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.5.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.6 CONTEX

The CONTEX subprogram is the standard or maximum likelihood classifier.

11.6.1 LINKAGES

The CONTEX subprogram calls the FDLINT subprogram. It is called by CLSFY2.

11.6.2 INTERFACES

The CONTEX subprogram interfaces with other routines through common block CLASS and through the calling arguments.

11.6.3 INPUTS

Calling sequence: CALL CONTEX(NCHAN,NC,NPTS,AVE,COV,BMATR,IDATA,VERTCS,VT,IR,VR,ILINE,TH)

Parameter	Dimension	In/out	Definition
NCHAN	1	In	Number of channels used in processing.
NC	1	In	Number of subclasses (training classes for which covariance matrices, mean vectors, and class-pair thresholds are available).
NPTS	1	In	Number of points in the rectangular field.
AVE	1	In	Array containing means.
COV	1	In	Array containing covariances.
BMATR	BMCOMB,BMFEAT	In	Array containing B-matrix, if available.

Parameter	Dimension	In/out	Definition
IDATA	1	In	Scan line of data to be classified.
VERTCS	1	In/out	Array containing vertices of the field to be classified.
VT	1	In/out	Number of vertices in the field to be classified.
IR	1	Out	Array containing classified data.
VR	1	Out	Corresponding PDF of IR.
ILINE	1	In/out	Scan line number.
TH	1	In	Precomputed class-pair thresholds.

11.6.4 OUTPUTS

The results are returned for use by the calling routine.

11.6.5 STORAGE REQUIREMENTS

This subprogram requires 3982 bytes of storage.

11.6.6 DESCRIPTION

This subprogram uses the modified Cholesky decomposition of the covariance matrix to compute the PDF. It scans the set of training classes and obtains the maximum probability (likelihood) over all classes for assigning a class to each pixel of the input scan line. The precomputed class-pair thresholds in TH are used to minimize the number of computed class PDF's to obtain the maximum PDF for the sample being classified. The class number and PDF for each sample on the input scan line are returned in the IR and VR arrays.

11.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.6.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.7 FALSY

The FALSY subprogram finds a root of the function G used in computing class-pair thresholds.

11.7.1 LINKAGES

The FALSY subprogram calls the G function. It is called by subprogram THRESH.

11.7.2 INTERFACES

The FALSY subprogram interfaces with other routines through the calling arguments.

11.7.3 INPUTS

Calling sequence: CALL FALSY(XL,XU,C,FXL,FXU,KC,XN,KT,T,K,KP1, S1,S2,U1,U2,BB)

Parameter	Dimension	In/out	Definition
XL	1	In/out	Lowest value for X in the interval.
XU	1	In/out	Highest value for X in the interval.
C	1	In	Value of the function evaluated at some point within the interval XL to XU.
FXL	1	Out	Lowest value for F(X).
FXU	1	Out	Highest value for F(X).
KC	1	In	Flag indicating whether or not to compute new values for F(XL) and F(XU).
XN	1	In/out	A possible root of the approximating quadratic use as a trial solution.

Parameter	Dimension	In/out	Definition
KT	1	In/out	Switch; if = 1, T is computed.
T	1	In/out	Threshold value.
K	1	In/out	Number of means for the class.
KP1	1	In/out	Number of features + 1.
S1	K,K	In/out	Array containing covariance matrix for class 1.
S2	K,K	In/out	Array containing covariance matrix for class 2.
U1	K	In	Array containing mean vector for class 1 of the class pair.
U2	K	In	Array containing mean vector for class 2 of the class pair.
BB	K,KP1	In/out	Working storage array.

11.7.4 OUTPUTS

The results are returned for use by the calling routine.

11.7.5 STORAGE REQUIREMENTS

This subprogram requires 2502 bytes of storage.

11.7.6 DESCRIPTION

The FALSY subprogram fits a quadratic to three points, XL, X, and XU, which produce an approximation of a function G[H(X)] within a defined interval. A root of the approximating quadratic, XN, is returned for use as a trial solution of G[H(X)] = C2 - C1.

11.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.7.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.8 G

The function G is a polynomial from which the class-pair threshold is computed.

11.8.1 LINKAGES

The function G calls the GJR subprogram. It is called by the THRESH and FALSY subprograms.

11.8.2 INTERFACES

The G subprogram interfaces with other routines through the calling arguments.

11.8.3 INPUTS

Calling sequence: G(A,S1,S2,U1,U2,BB,KT,T,K,KP1)

Parameter	Dimension	In/out	Definition
A	1	In	Point at which the polynomial G is evaluated.
S1	K,K	In	Array containing covariance matrix for class 1 of the class pair.
S2	K,K	In	Array containing covariance matrix for class 2 of the class pair.
U1	K	In	Mean vector for class 1.
U2	K	In	Mean vector for class 2.
BB	K,KP1	Out	Working storage.
KT	1	In	Switch; if = 1, T is computed.
T	1	Out	Threshold value.
K	1	In	Number of means for the class.
KP1	1	In/out	Number of features + 1.

11.8.4 OUTPUTS

The results are returned for use by the calling routine.

11.8.5 STORAGE REQUIREMENTS

This subprogram requires 1884 bytes of storage.

11.8.6 DESCRIPTION

Not required.

11.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.8.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.9 GJR

The GJR function is used to calculate the polynomial G.

11.9.1 LINKAGES

The GJR function does not call any other subprogram. It is called by the function G.

11.9.2 INTERFACES

The GJR function interfaces with other routines through the calling arguments.

11.9.3 INPUTS

Calling sequence: CALL GJR(A,NC,NR,N,MC,*,JC,V)

Parameter	Dimension	In/out	Definition
A	NR,NC	In/out	Working storage.
NC	1	In	Number of features + 1.
NR	1	In	Number of means for the class.
N	1	In	Number of means for the class.
MC	1	In	Number of features + 1.
*	1	Out	Multiple return parameter.
JC	1	Out	Indexing array.
V	2	Out	Storage vector.

11.9.4 OUTPUTS

The results are returned for use by the calling routine.

11.9.5 STORAGE REQUIREMENTS

This subprogram requires 2756 bytes of storage.

11.9.6 DESCRIPTION

Not required.

11.9.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.9.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.10 MAPHDG

The MAPHDG subprogram prints the header information for the classification map produced by the CLASSIFY processor.

11.10.1 LINKAGES

The MAPHDG subroutine does not call any other subprogram. It is called by CLSFY2.

11.10.2 INTERFACES

The MAPHDG subprogram interfaces with other routines through common block INFORM and through the calling arguments.

11.10.3 INPUTS

Calling sequence: CALL MAPHDG(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX,SUBNO,SUBDES)

Parameter	Dimension	In/out	Definition
NOCAT	1	In	Number of categories.
CLSSYM	1	In	Symbols for categories or subclasses.
CATNAM	1	In	Category names.
KATNO	1	In	Category-class correspondence array (category to which each class was assigned).
CLSMTX	1	In	Matrix of training class names.
SUBNO	1	In	Array containing subclass numbers in each class.
SUBDES	1	In	Array of subclass names.

11.10.4 OUTPUTS

This subprogram prints a map of classification results.

11.10.5 STORAGE REQUIREMENTS

This subprogram requires 1860 bytes of storage.

11.10.6 DESCRIPTION

The MAPHDG subprogram accepts classification information from the calling routine via input arguments. If categories have been input, it prints category classifier information, including category name, class name, subclass name and number, and the symbol for each category or subclass. After it prints category classifier information, the subprogram prints standard classifier information, including class and subclass name, number, and symbol. If categories are not identified in the input, only the standard classifier information is printed.

11.10.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.10.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.11 MCHLSK

The MCHLSK subprogram computes the modified Cholesky decomposition of the covariance matrix. The decompositions overlay the elements of the covariance matrix.

11.11.1 LINKAGES

The MCHLSK subroutine does not call any other subprogram. It is called by CLSFY1.

11.11.2 INTERFACES

The MCHLSK subprogram interfaces with other routines through the calling arguments.

11.11.3 INPUTS

Calling sequence: CALL MCHLSK(KK,NV,DUM,DET)

Parameter	Dimension	In/out	Definition
KK	1	In/out	Covariance matrix in symmetric storage.
NV	1	In	Number of channels.
DUM	1*	Out	A work area of size NV - 1.
DET	1	Out	Determinant of the covariance matrix.

11.11.4 OUTPUTS

The results are stored for use by the calling routine.

11.11.5 STORAGE REQUIREMENTS

This subprogram requires 2204 bytes of storage.

*Double precision.

11.11.6 DESCRIPTION

The MCHLSK subprogram sets up a loop over all input channels and (1) computes the elements of the diagonal matrix D and stores them in the diagonal elements of the covariance matrix KK, (2) computes the elements of the lower triangular matrix L and stores them in the off-diagonal elements of KK, and (3) obtains the modified Cholesky factorization of the input matrix and stores the result in symmetric notation in KK.

11.11.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.11.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.12 REDIF2

The REDIF2 subprogram reads and analyzes supervisor control cards.

11.12.1 LINKAGES

The REDIF2 subprogram calls the BMFIL, CATSCN, CMERR, CRDSTA, FIND12, FLTNUM, GRPSCN, NUMBER, NXTCHR, and ORDER subprograms. It is called by SETUP2.

11.12.2 INTERFACES

The REDIF2 subprogram interfaces with other routines through common blocks CLASS, GLOBAL, and INFORM and through the calling arguments.

11.12.3 INPUTS

Calling sequence: CALL REDIF2(ARRAY,TOP,APRIOR,KATNO,BMATRX, PRIORI)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
APRIOR	1	In	A priori probability values for each class; initialized to 0.0 in REDIF2.
KATNO	60	In	Category-class correspondence array (unused in REDIF2).
BMATRX	BMCOMB,BMFEAT	Out	Storage for the B-matrix, if available.

Parameter	Dimension	In/out	Definition
PRIORI	60	Out	Storage for a priori probability values read from input APRIOR control card.

The control cards relevant to this routine are input to the calling routine, SETUP2.

11.12.4 OUTPUTS

This subprogram outputs results on punched cards, tape, and line printer.

11.12.5 STORAGE REQUIREMENTS

This subprogram requires 6038 bytes of storage.

11.12.6 DESCRIPTION

The REDIF2 subprogram reads control card images and determines the type of information to be input; i.e., subclass, channel, date, comment, heading, option, group, B-matrix, a priori probability values, category, data file, or statistics file input. If an error is detected on a card image, a message is generated and the next card is read. If no error is detected, the card content is written on the specified unit. The date, comment, and heading cards are read, and this information is printed. Subprogram ORDER is called to arrange SUBCLASS and CHANNELS cards in ascending order. If the B-matrix is input, the BMFIL subprogram is called to read and write the B-matrix file. If an error is detected, program execution terminates from REDIF2.

If no error has occurred, REDIF2 reads the APRIOR card, which indicates the input of training class a priori probability values. If an error is detected, a message is generated and default a priori probability values are used.

11.12.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.12.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.13 RELERR

The RELERR subprogram is a function which computes the Euclidean norm of the covariance matrix before the Cholesky factorization.

11.13.1 LINKAGES

The RELERR function does not call any other subprogram. It is called by CLSFY1.

11.13.2 INTERFACES

The RELERR subprogram interfaces with other routines through common block SCRACH and through the calling arguments.

11.13.3 INPUTS

Calling sequence: CALL RELERR(COVMTX,COV,NOFET2,VARSZ2)

Parameter	Dimension	In/out	Definition
COVMTX	VARSZ2	In/out	Array containing covariance matrices (symmetric storage) for NOCLS2 training classes.
COV	VARSZ2	In	Array of covariances.
NOFET2	1	In	Number of channels to be used for processing.
VARSZ2	1	In	Size of each covariance matrix; $\text{NOFET2} \times \frac{\text{NOFET2 + 1}}{2}$.

11.13.4 OUTPUTS

The results are returned for use by the calling routine.

11.13.5 STORAGE REQUIREMENTS

This subprogram requires 1372 bytes of storage.

11.13.6 DESCRIPTION

Not required.

11.13.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.13.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.14 SETUP2

Subprogram SETUP2 analyzes supervisory information and reduces statistics for processing.

11.14.1 LINKAGES

The SETUP2 subroutine calls the CMERR, FSBSFL, LAREAD, REDIF2, REDSAV, and WRTBMT subprograms. It is called by the CLSFY driver routine.

11.14.2 INTERFACES

The SETUP2 subprogram interfaces with other routines through common blocks CLASS, GLOBAL, and INFORM and through the calling arguments.

11.14.3 INPUTS

Calling sequence: CALL SETUP2(ARRAY,TOP,FLDFLG,APRIOR,BMATRX, KATNO)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
FLDFLG	1	In	Unused in SETUP2.
APRIOR	1	Out	Array containing a priori probability values for each subclass.
BMATRX	BMCOMB,BMFEAT	Out	Storage array for the B-matrix.
KATNO	60	Out	Class-category correspondence array.

The control cards relevant to this routine are given in section 11.5.3 (table 11-1) of volume II.

11.14.4 OUTPUTS

The results are written on the MAPTAP file.

11.14.5 STORAGE REQUIREMENTS

This subprogram requires 7740 bytes of storage.

11.14.6 DESCRIPTION

The SETUP2 subprogram positions the MAPTAP file for an unformatted write and calls REDIF2 to read supervisory information from control cards and REDSAV to read the SAVTAP file and reduce statistics. It tests for category file information and, if available, forms category/class/subclass associations using class/subclass information. These data are stored in arrays CATNAM and KCLSNA. If channels are not input, the same channels selected from the SAVTAP file are used.

SETUP2 computes base addresses for class-pair thresholds and for data returned from the TAPHDR subprogram. It stores blanks in default symbols for use in printing the map classification for the unclassified pixels. It assigns available classes to categories and stores them in the newly created KATNO array. Subclasses are assigned to categories, and the SUBCAT array is set up to contain the category to which each class belongs. Supervisory information is printed, including file numbers and the total numbers of classes, subclasses, fields, and channels.

If user-input a priori values do not equal 1.0, they are normalized by SETUP2, which writes the values according to their input; i.e., by class, subclass, or category. The program will ignore user-input values if an error occurs in the APRIOR control card. In the absence of user input, it computes default values for either the category classifier or the standard classifier or from the SAVTAP file.

Results from SETUP2 are written as header record 1 of the MAPTAP file. If NOCLS2 $\leq$ 1, the LAREAD function is invoked to read field definition card images. If NOCLS2 $\geq$ 1, control is returned to the calling routine.

11.14.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.14.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.15 SETUS

The SETUS subprogram uses the lower triangular matrix and the diagonal matrix for a specific class to compute the full covariance matrix for the class. It also extracts the mean vector for the class and returns it in the vector MEAN.

11.15.1 LINKAGES

The SETUS subroutine does not call any other subprogram. It is called by the THRESH subprogram.

11.15.2 INTERFACES

The SETUS subprogram interfaces with other routines through the calling arguments.

11.15.3 INPUTS

Calling sequence: CALL SETUS(AVEMTX,COVMTX,DIAG,MEAN,COV,NCHAN, NOCLS2,VARSZ2,JJ)

Parameter	Dimension	In/out	Definition
AVEMTX	NCHAN,NOCLS2	In	Array of NOCLS2 training class mean vectors (NOFET2 means per class).
COVMTX	VARSZ2,NOCLS2	In/out	Array of covariance matrices (symmetric storage) for NOCLS2 training classes.
DIAG	NCHAN	Out	Array containing diagonal matrix for class JJ in COVMTX.
MEAN	NCHAN	Out	Array containing mean vector for class JJ in AVEMTX.

Parameter	Dimension	In/out	Definition
COV	NCHAN,NCHAN	Out	Array of covariances.
NCHAN	1	In	Number of channels used in processing.
NOCLS2	1	In	Number of classes from the SAVTAP file to be processed.
VARSZ2	1	In	Size of each covariance matrix; NOFET2 $\times \frac{\text{NOFET2} + 1}{2}$.
JJ	1	In	Class being processed.

11.15.4 OUTPUTS

The results are returned for use by the calling routine.

11.15.5 STORAGE REQUIREMENTS

This subprogram requires 1270 bytes of storage.

11.15.6 DESCRIPTION

Not required.

11.15.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.15.8 LISTING

The subprogram listing is provided in volume IV, section 11.

11.16 THRESH

The THRESH subprogram computes class-pair thresholds.

11.16.1 LINKAGES

The THRESH subroutine calls the FALSY, G, and SETUS subprograms. It is called by CLSFY1.

11.16.2 INTERFACES

The THRESH subprogram interfaces with other routines through the calling arguments.

11.16.3 INPUTS

Calling sequence: CALL THRESH(NOCLS2,NOFET2,NPL1,APRIOR,AVEMTX, COVMTX,DET,VARSZ2,S1,S2,U1,U2,BB,DIAG,THIJ)

Parameter	Dimension	In/out	Definition
NOCLS2	1	In	Number of classes from the SAVTAP file to be used for processing.
NOFET2	1	In	Number of channels to be used for processing.
NPL1	1	In	Number of features + 1.
APRIOR	NOCLS2	In	Array containing a priori probability values for each class.
AVEMTX	NOFET2,NOCLS2	In	Array of NOCLS2 training class mean vectors (NOFET2 means per class).
COVMTX	VARSZ2,NOCLS2	In	Array of covariance matrices (symmetric storage) for NOCLS2 training classes.

Parameter	Dimension	In/out	Definition
DET	NOCLS2	In	Array of subclass determinants.
VARSZ2	1	In	Size of each covariance matrix; $\text{NOFET2} \times \frac{\text{NOFET2} + 1}{2}$.
S1	NOFET2,NOFET2	In	Array containing the covariance matrix for class 1.
S2	NOFET2,NOFET2	In	Array containing the covariance matrix for class 2.
U1	NOFET2	In	Array containing the mean vector for class 1 of the class pair.
U2	NOFET2	In	Array containing the mean vector for class 2 of the class pair.
BB	NOFET2,NPL1	In	Working storage array.
DIAG	NOFET2	In	Array containing diagonal matrix of classes.
THIJ	1	Out	Array for symmetric storage of computed class-pair thresholds.

11.16.4 OUTPUTS

This subprogram outputs the class-pair threshold array on the line printer.

11.16.5 STORAGE REQUIREMENTS

This subprogram requires 2260 bytes of storage.

11.16.6 DESCRIPTION

Subprogram THRESH calls SETUS for the full covariance matrix and mean vector for the class. It computes the symmetric matrix storage location for the class I-J pair threshold value and uses subprogram FALSY and function G to compute the class-pair thresholds. The computed class-pair threshold values are stored in THIJ.

11.16.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 11.17.

11.16.8 LISTING

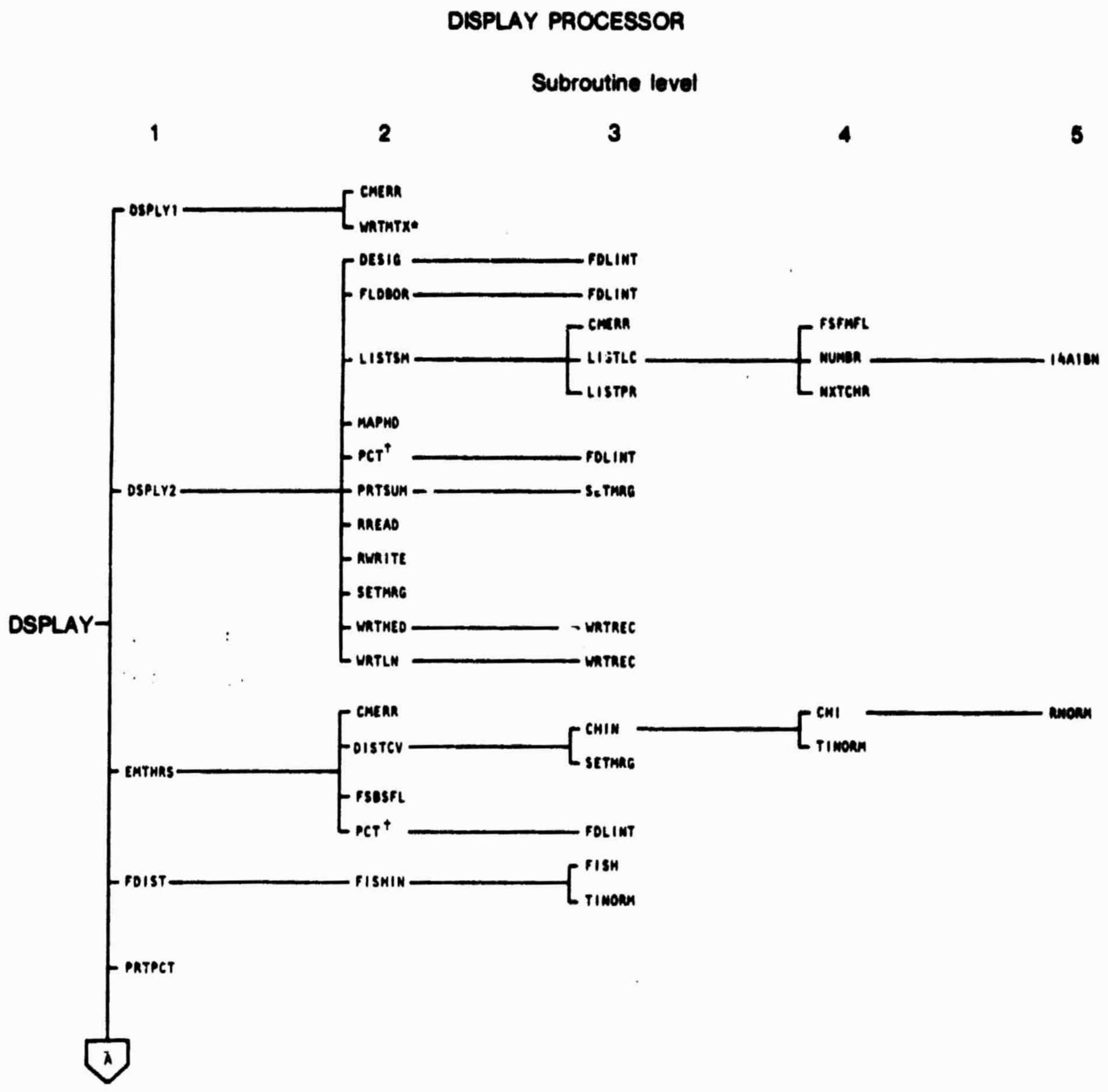
The subprogram listing is provided in volume IV, section 11.

11.17 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the CLASSIFY processor.

12. DISPLAY PROCESSOR SUBPROGRAMS

The DISPLAY processor reads the MAPTAP file output by the CLASSIFY processor, performs various operations in the classification process, and displays classification results on tape and line printer. The DISPLAY processor has 22 subprograms within the processor and, in addition, invokes 20 utility subprograms during execution. Figure 12-1 is a linkage diagram of the DISPLAY processor.



*Entry point is DWRTMX.
†Entry point is PCTT.

Figure 12-1.- Linkage diagram for the DISPLAY processor.

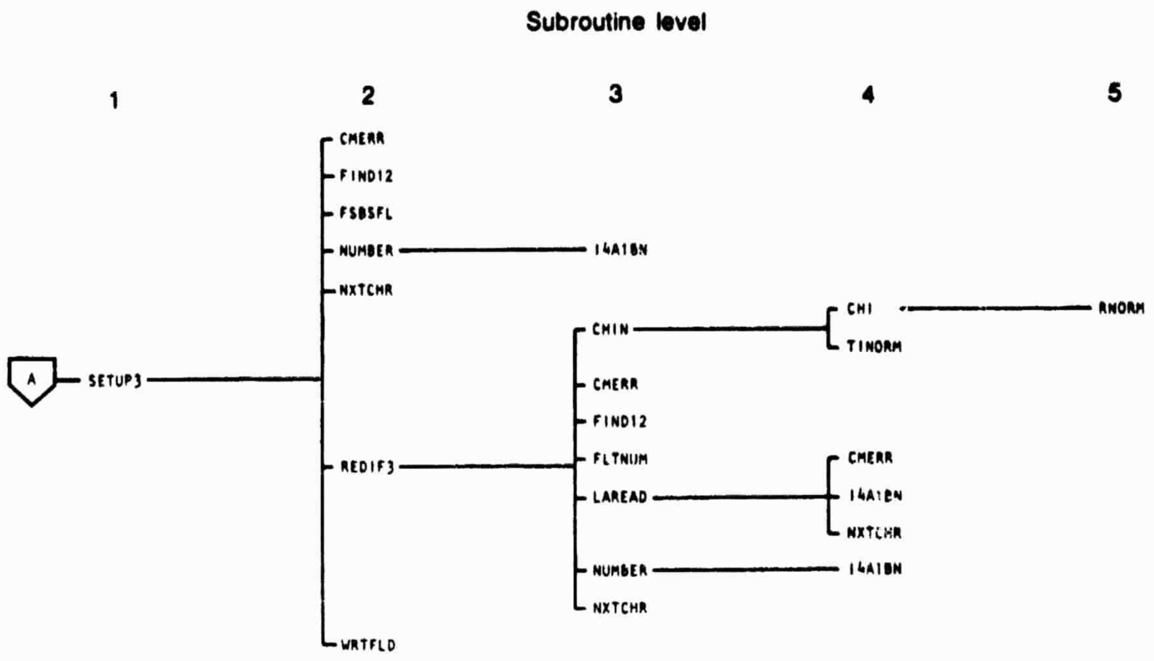


Figure 12-1.- Concluded.

12.1 DSPLAY

The DSPLAY subprogram is the driver routine for the DISPLAY processor, which displays the classification results obtained by the CLASSIFY and LABEL processors.

12.1.1 LINKAGES

The DSPLAY subprogram calls the DSPLY1, DSPLY2, EMTHRS, FDIST, PRTPCT, and SETUP3 subprograms. It is called by MONTOR.

12.1.2 INTERFACES

The DSPLAY subprogram interfaces with other routines through common block DISPL and through the calling arguments.

12.1.3 INPUTS

Calling sequence: CALL DSPLAY(ARRAY,TOP)

Parameter	Dimension	In/out	Definition
ARRAY	1	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable location in array; TOP = 10 600.

12.1.4 OUTPUTS

Not applicable.

12.1.5 STORAGE REQUIREMENTS

This subprogram requires 1388 bytes of storage.

12.1.6 DESCRIPTION

The DSPLAY routine coordinates the subprograms which read the MAPTAP file and display classification results. Calls are made to the following subprograms from DSPLAY.

a. SETUP3 — to read the MAPTAP and control cards and store training and/or field definitions in ARRAY.

b. DSPLY1 — to read the next two records from the MAPTAP file and print statistics if requested.

c. EMTHRS — to compute and plot a histogram of the quadratic form for correctly classified pixels in training or test fields.

d. FDIST — to compute and store Fisher F-distribution thresholds.

e. DSPLY2 — to print a classification map and build classification performance tables.

f. PRTPCT — to print classification performance tables.

12.1.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.1.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.2 CHI

The CHI subprogram is a function that computes the chi-squared distribution with N degrees of freedom.

12.2.1 LINKAGES

The CHI function calls the RNORM function. It is called by the CHIN subprogram.

12.2.2 INTERFACES

The CHI function interfaces with other routines through the calling arguments.

12.2.3 INPUTS

Calling sequence: CHI(X,N,IFLAG)

Parameter	Dimension	In/out	Definition
X	1	In	Input threshold value; if > 0, the chi-squared distribution may be computed; otherwise, control is returned to the calling routine.
N	1	In	Number of degrees of freedom.
IFLAG	1	Out	Flag indicating CHI has been calculated; if IFLAG = 1, the threshold value is outside the allowable range.

12.2.4 OUTPUTS

The results are returned for use by the calling routine.

12.2.5 STORAGE REQUIREMENTS

This subprogram requires 898 bytes of storage.

12.2.6 DESCRIPTION

The CHI function calculates the CHI distribution for one, two, or more degrees of freedom if the input parameter X is a positive real number. Degrees of freedom are checked to ascertain if the number of degrees is even or odd; if odd, CHI is calculated for one and, if even, for two degrees of freedom. If $N \geq 3$, CHI is calculated for N degrees of freedom. If an overflow occurs, the function returns a value of 1 for IFLAG.

12.2.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.2.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.3 CHIN

The CHIN subprogram is a function that calculates the chi-squared threshold values.

12.3.1 LINKAGES

The CHIN function calls the CHI and TINORM functions. It is called by the DISTCV and REDIF3 subprograms.

12.3.2 INTERFACES

The CHIN function interfaces with other routines through the calling arguments.

12.3.3 INPUTS

Calling sequence: CHIN(ALPHA,N,IFLAG)

Parameter	Dimension	In/out	Definition
ALPHA	1	In	Array containing input confidence levels. (Confidence level is defined to be the ratio of the number of dots within the given analyst-labeled category to the total number of dots classified into each category.)
N	1	In	Number of degrees of freedom.
IFLAG	1	In/out	Flag indicating that CHI has been calculated; if IFLAG = 1, the threshold value is outside the allowable range.

12.3.4 OUTPUTS

The results are returned for use by the calling routine.

12.3.5 STORAGE REQUIREMENTS

This subprogram requires 1866 bytes of storage.

12.3.6 DESCRIPTION

Not required.

12.3.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.3.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.4 DESIG

The DESIG subprogram sets the IR array for DO/DU fields and locates DO/DU fields on each line.

12.4.1 LINKAGES

The DESIG subprogram calls the FDLINT subprogram. It is called by DSPLY2.

12.4.2 INTERFACES

The DESIG subprogram interfaces with other routines through the calling arguments.

12.4.3 INPUTS

Calling sequence: CALL DESIG(LINE,IR,FLDSAV,FIELD,VERTEX,NOFLD, SAMSTR,SAMEND,SAMINC)

Parameter	Dimension	In/out	Definition
LINE	1	In	Number of line being scanned.
IR	1	Out	Array containing subclass numbers to which points were assigned on the line being processed.
FLDSAV	4,NOFLD		Array containing field information, including field name, class and subclass numbers, and number of vertices; FLDSV2 = training fields; FLDSV3 = test fields.
FIELD	5,NOFLD	In	Array containing coordinates of the rectangular area surrounding each field, including line start and end, sample start and end, and pointer into VERTEX array for

Parameter	Dimension	In/out	Definition
			field vertices; FIELD2 = training field locations; FIELD3 = test field locations.
VERTEX	1	In	Array containing field vertices; VERTX2 = training field vertices; VERTX3 = test field vertices.
NOFLD	1	In	Number of fields to process; NOFLD2 = training fields; NOFLD3 = test fields.
SAMSTR	1	In	Beginning sample number for classified field.
SAMEND	1	In	Last sample number for classified field.
SAMINC	1	In	Sample increment used by the CLASSIFY processor.

12.4.4 OUTPUTS

The results are returned for use by the calling routine.

12.4.5 STORAGE REQUIREMENTS

This subprogram requires 1234 bytes of storage.

12.4.6 DESCRIPTION

Not required.

12.4.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.4.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.5 DISTCV

The DISTCV subprogram plots the distribution and chi-squared curves and computes empirical threshold values.

12.5.1 LINKAGES

The DISTCV subprogram calls the CHIN and SETMRG subprograms. It is called by EMTHRS.

12.5.2 INTERFACES

The DISTCV subprogram interfaces with other routines through common block DISPL and through the calling arguments.

12.5.3 INPUTS

Calling sequence: CALL DISTCV(DSFUNC,TOTPTS,RANGE)

Parameter	Dimension	In/out	Definition
DSFUNC	RANGE,60	In	Array containing the histogram of the quadratic form.
TOTPTS	1	Out	Array containing total histogrammed pixels for each subclass.
RANGE	1	In	Range of the histogram.

12.5.4 OUTPUTS

This subprogram outputs results on the line printer or other specified unit.

12.5.5 STORAGE REQUIREMENTS

This subprogram requires 4404 bytes of storage.

12.5.6 DESCRIPTION

The DISTCV subprogram calls function CHIN to calculate the chi-squared distribution threshold values; plots the chi-squared and class distribution curves; and outputs the chi-squared thresholds, empirical thresholds, and the user rejection percentages for each subclass.

12.5.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.5.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.6 DSPLY1

The DSPLY1 subprogram reads, writes, and stores the statistics from the MAPTAP file.

12.6.1 LINKAGES

The DSPLY1 subprogram calls the CMERR and WRTMTX subprograms. It is called by the DSPLAY driver routine.

12.6.2 INTERFACES

The DSPLY1 subprogram interfaces with other routines through common blocks DISPL and GLOBAL.

12.6.3 INPUTS

Input to the DSPLY1 subprogram consists of the MAPTAP or MAPUNT file output by the CLASSIFY or LABEL processor.

Calling sequence: CALL DSPLY1

12.6.4 OUTPUTS

This subprogram outputs the mean vectors, covariance matrices, PDF constants, and determinants on the line printer or other specified unit.

12.6.5 STORAGE REQUIREMENTS

This subprogram requires 38 778 bytes of storage.

12.6.6 DESCRIPTION

Utilizing internal subroutine DSPL1A, subprogram DSPLY1 reads (from the MAPTAP file) the original mean vector and covariance matrix and the transformed B-matrix (if applied by the CLASSIFY processor) for NOCLS2 classes and NOFET2 channels. It reads the covariance matrix (after Cholesky factorization), the PDF constants, and the determinant for the covariance matrix for each class. Utility subprogram WRTMTX is called to print the covariance matrix. DSPL1A prints all other retrieved statistics. If storage is insufficient for the covariance matrix, DSPLY1 generates a message and exits via subprogram CMERR.

12.6.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.6.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.7 DSPLY2

The DSPLY2 subprogram performs spatial filtering and thresholding on the classified data, prints the classification map, and calls the appropriate routines to build and print classification performance tables.

12.7.1 LINKAGES

The DSPLY2 subprogram calls the DESIG, FLDBOR, LISTSM, MAPHD, PCT, PRTSUM, RREAD, RWRITE, SETMRG, WRTHED, and WRTLN subprograms. It is called by the DSPLAY driver routine.

12.7.2 INTERFACES

The DSPLY2 subprogram interfaces with other routines through common blocks DISPL and GLOBAL and through the calling arguments.

12.7.3 INPUTS

Input to the DSPLY2 subprogram consists of the MAPTAP or MAPUNT file output by the CLASSIFY or LABEL processor.

Calling sequence: CALL DSPLY2(TRNSAV,TRNFLD,TRNVER,TSTSAV,TSTFLD, TSTVER,PCTAB,GTUNIT,GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,NAMECT,ALP, DESSAV,DESFLD,DESVER,NOFLD4)

Parameter	Dimension	In/out	Definition
TRNSAV	4,NOFLD2	In	Array initialized in SETUP2 as follows: If DOTKEY > 0: TRNSAV(1,I) = unused (scratch); TRNSAV(2,I) = number for the MAPTAP category matching the dot category name; TRNSAV(3,I) = dot type (1 or 2); TRNSAV(4,I) = number of vertices for dot I, I = 1,2,···,NOFLD2. (NOFLD2 = total number of dots.)

Parameter	Dimension	In/out	Definition
TRNFLD	5,NOFLD2	In	Array containing coordinates of rectangular area surrounding each dot; initialized in SETUP3 as follows: TRNFLD(1,I) = starting line number; TRNFLD(2,I) = ending line number; TRNFLD(3,I) = starting sample number; TRNFLD(4,I) = ending sample number; and TRNFLD(5,I) = location in TRNVER for vertex of dot I; I = 1,2,···,NOFLD2.
TRNVER	2,TOTVT2	In	Array storage for training field (dot) vertex coordinates: TRNVER(1,I) = sample number; TRNVER(2,I) = line number; I = 1,2,···, TOTVT2. (TOTVT2 = total number of vertices for NOFLD2 dots.)
TSTSAV	4,NOFLD3	In	Array storage for test or designated fields; equivalent to TRNSAV for training fields. [NOFLD3 = total number of test or designated fields ($\leq$200).]
TSTFLD	5,NOFLD3	In	Array for storage of test or designated fields; equivalent to TRNFLD for training fields.

Parameter	Dimension	In/out	Definition
TSTVER	2,TOTVT3	In	Array for storage of test or designated fields; information equivalent to TRNVER for training fields. (TOTVT3 = total number of vertices for all test or designated fields.)
PCTAB	NOTRFD,NOSUB3	In	Array containing classification performance tabulation.
GTUNIT	1	In/out	Number of unit containing ground-truth dot data.
GTFILE	1	In/out	Number of the GTUNIT file to be processed.
AIUNIT	1	In/out	Number of unit containing the AI dot data file.
AIFILE	1	In/out	Number of the AIUNIT file to be processed.
PPUNIT	1	In/out	Number of the unit containing the PPTC dot data.
PPFILE	1	In/out	Number of the PPUNIT file to be processed.
NAMECT	1	In/out	Name of small grains category which has been invoked by the category classifier.
ALP	2	In/out	Array containing floating-point values of α_1 and α_2 for bias correction.
DESSAV	4,50	In/out	Array containing DO/DU field information, including field name, class and subclass numbers, and number of vertices.

Parameter	Dimension	In/out	Definition
DESFLD	5,50	In/out	Array containing coordinates of the rectangular area surrounding each DO/DU field, including line start and end and pointer into DESVER array for field vertices.
DESVER	1100	In/out	Array containing DO/DU field vertex coordinates.
NOFLD4	1	In/out	Number of DO/DU fields being processed.

12.7.4 OUTPUTS

This subprogram outputs the classification performance tables, classification map, and ITS summary report on the line printer.

12.7.5 STORAGE REQUIREMENTS

This subprogram requires 67 542 bytes of storage.

12.7.6 DESCRIPTION

The DSPLY2 subprogram is the prime routine for displaying classification results. Two main modes of operation are characteristic of DSPLY2 — the dot processing mode and processing without dot data. In both modes, DSPLY2 prints headings and column numbers, performs thresholding and spatial filtering, and writes performance summaries.

On a field-by-field basis, DSPLY2 reads the field header record from the MAPTAP file and builds summary tables using classification information for every classified pixel. The MAPTAP file is read, one line at a time, and field information (name, class and subclass numbers, and number of vertices), rectangular coordinates, and the likelihood ratio of the pixel are placed on a buffer. (Lines are stored three at a time for spatial filtering.) Line number = 0 indicates the end of the field, in which case the next field will be read for processing.

A test is performed for designated fields according to the value of the parameter DESKEY. If DESKEY = 1, subprogram DESIG is called to locate DO/DU fields on each line and to set the IR array for the designated pixels.

Thresholding is performed according to the value of each pixel in the IR array. If = 0, no thresholding is performed; if $\leq$ NOSUB3, thresholding is performed and the pixel is added to the thresholded class and total thresholded pixels; and, if $>$ NOSUB3, the pixel is placed in the DO/DU category. The program adds the total number of pixels in the field.

Spatial filtering (on each group of three lines stored on the buffer) is performed on option using the four-nearest-neighbors procedure. This can change the subclass assignment of a pixel.

A check is made for the dot or nondot processing mode. If non-dot mode, subprogram PCT is called to build the classification performance table (PCTAB); subprogram PRTPCT to print the PCTAB; subprogram FLDBOR to place a border around test or training fields on the classification map; and subprogram PRTSUM to print the classification summary for each field and, optionally, the ITS summary report.

In the dot processing mode, internal subprogram DOTPCT performs dot processing. DOTPCT is called once for each classified scan line input from MAPTAP in DSPLY2. After the test is made in DSPLY2 to determine that dots are present within the given scan line number, a check is made on the pixel number for each dot to ascertain that the dot is present in the actual pixels classified. For those dots present in the given classification record (IR), the subclass number at the dot's position in the classification record is extracted from IR and placed into the dot's position in PCTAB. For those dots having pixel coordinates which are not in the classification record, the subclass number is placed in PCTAB as a zero. A loop is used to check dots and initialize PCTAB. The classification performance for these dots is returned in the argument PCTAB.

Subprogram LISTSM is called to generate dot performance summary reports.

12.7.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.7.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.8 EMTHRS

The EMTHRS subprogram calculates empirical thresholds when this option is exercised and plots histograms of the quadratic form for correctly classified pixels within training or test fields.

12.8.1 LINKAGES

The EMTHRS subprogram calls the CMERR, DISTCV, FSBSFL, and PCT subprograms. It is called by the DSPLAY driver routine.

12.8.2 INTERFACES

The EMTHRS subprogram interfaces with other routines through common blocks DISPL and GLOBAL and through the calling arguments.

12.8.3 INPUTS

Input to the EMTHRS subprogram consists of the MAPTAP file output by the CLASSIFY processor.

Calling sequence: CALL EMTHRS(FLDSAV,FIELD,VERTEX,NOFLD)

Parameter	Dimension	In/out	Definition
FLDSAV	1	In/out	Address in ARRAY where field information is stored, including field name, class and subclass numbers, and number of vertices; FLDSV2 = training field information; FLDSV3 = test field information.
FIELD	1	In/out	Address in ARRAY where coordinates of the rectangular area surrounding each field are stored, including line start and end, sample start and end, and pointer into VERTEX

Parameter	Dimension	In/out	Definition
			array for field vertices; FIELD2 = training field locations; FIELD3 = test field locations.
VERTEX	1	In/out	Address in ARRAY for field vertices; VERTX2 = training field vertices; VERTX3 = test field vertices.
NOFLD	1	In/out	Number of fields used in processing; NOFLD2 = training fields; NOFLD3 = test fields.

12.8.4 OUTPUTS

The results are returned for use by the calling routine.

12.8.5 STORAGE REQUIREMENTS

This subprogram requires 33 144 bytes of storage.

12.8.6 DESCRIPTION

The EMTHRS subprogram reads field information from the MAPTAP file, including points, lines, and field descriptions. If PTS ≠ 0, it retrieves subclass numbers for each pixel from the classified array IR and corresponding PDF's from the array VR and calls PCTT to produce a histogram of the quadratic form within ground-truth fields. This process continues until all fields of MAPTAP address ILINE are processed. If PTS = 0, or after all fields are processed, EMTHRS calls subprogram DISTCV to plot the distribution and chi-squared curves and compute empirical threshold values.

12.8.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.8.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.9 FDIST

The FDIST subprogram coordinates the function used to obtain the Fisher F-distribution threshold values.

12.9.1 LINKAGES

The FDIST subprogram calls the FISHIN function. It is called by the DSPLAY driver routine.

12.9.2 INTERFACES

The FDIST subprogram interfaces with other routines through common block DISPL.

12.9.3 INPUTS

Calling sequence: CALL FDIST

12.9.4 OUTPUTS

The results are returned for use by the calling routine.

12.9.5 STORAGE REQUIREMENTS

This subprogram requires 1032 bytes of storage.

12.9.6 DESCRIPTION

The FDIST subprogram accepts confidence levels and numbers of subclasses, samples, pixels, and channels from common block DISPL as input and calls the FISHIN function to compute each subclass threshold value. If an overflow occurs in FISHIN, it sets the threshold value at 999.999 and generates a message. The program continues until all subclass thresholds have been calculated.

12.9.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.9.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.10 FISH

The FISH subprogram is a function used in calculating the Fisher F-distribution threshold values.

12.10.1 LINKAGES

This function does not call any other subprogram. It is called by FISHIN.

12.10.2 INTERFACES

The FISH function interfaces with other routines through the calling arguments.

12.10.3 INPUTS

Calling sequence: FISH(F,N1,N2)

Parameter	Dimension	In/out	Definition
F	1	In	Partially calculated threshold value passed from FISHIN.
N1	1	In/out	Number of channels used in classification.
N2	1	In/out	Number of samples.

12.10.4 OUTPUTS

The results are returned for use by the calling routine.

12.10.5 STORAGE REQUIREMENTS

This subprogram requires 1590 bytes of storage.

12.10.6 DESCRIPTION

Not required.

12.10.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.10.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.11 FISHIN

The FISHIN subprogram is a function used to compute the Fisher F-distribution thresholds.

12.11.1 LINKAGES

This routine calls the FISH and TINORM subprograms. It is called by FDIST.

12.11.2 INTERFACES

The FISHIN function interfaces with other routines through the calling arguments.

12.11.3 INPUTS

Calling sequence: FISHIN(ALPHA,N1,N2,IFLAG)

Parameter	Dimension	In/out	Definition
ALPHA	1	In	1 - THRES(I); THRES(I) contains the input confidence level for sub-class i.
N1	1	In	Number of channels used in classification.
N2	1	In	Number of samples.
IFLAG	1	Out	Flag to set threshold value at 999.999 if overflow occurs.

12.11.4 OUTPUTS

The results are returned for use by the calling routine.

12.11.5 STORAGE REQUIREMENTS

This subprogram requires 1864 bytes of storage.

12.11.6 DESCRIPTION

Not required.

12.11.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.11.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.12 FLDBOR

The FLDBOR subprogram sets the symbol index in the classified line array IR for outlining training or test fields on the classification map.

12.12.1 LINKAGES

The FLDBOR subprogram calls the FDLINT subprogram. It is called by DSPLY2.

12.12.2 INTERFACES

The FLDBOR subprogram interfaces with other routines through the calling arguments.

12.12.3 INPUTS

Calling sequence: CALL FLDBOR(ISYM,LINUM,IR,NOFLD,FIELD,FLDSAV, VERTEX,NOSUB3,SAMSTR,SAMEND,SAMINC,LININC)

Parameter	Dimension	In/out	Definition
ISYM	1	In	Symbol used to outline field.
LINUM	1	In	Line number of the classified line.
IR	1	Out	Array containing subclass numbers to which points or pixels were assigned on the line being processed.
NOFLD	1	In	Number of fields to process; NOFLD2 = training fields; NOFLD3 = test fields.

Parameter	Dimension	In/out	Definition
FIELD	5,NOFLD	In	Array containing coordinates of the rectangular area surrounding each field, including line start and end, sample start and end, and pointer into the VERTEX array for field vertices; FIELD2 = training field locations; FIELD3 = test field locations.
FLDSAV	4,NOFLD	In	Array containing field information, including field name, class and subclass numbers, and number of vertices; FLDSV2 = training field information; FLDSV3 = test field information.
VERTEX	1	In	Array containing field vertices; VERTX2 = training field vertices; VERTX3 = test field vertices.
NOSUB3	1	In	Number of subclasses used by the CLASSIFY processor; NOSUB3 + 1 is the threshold class.
SAMSTR	1	In	Beginning sample number for the classified field.
SAMEND	1	In	Ending sample number for the classified field.
SAMINC	1	In	Sample increment used by the CLASSIFY processor.
LININC	1	In	Line increment.

12.12.4 OUTPUTS

The results are returned for use by the calling routine.

12.12.5 STORAGE REQUIREMENTS

This subprogram requires 1904 bytes of storage.

12.12.6 DESCRIPTION

Not required.

12.12.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.12.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.13 LISTPR

The LISTPR subprogram prints three classification performance tables of dot data:

a. PPTC labels versus classified labels

b. GT labels versus classified labels

c. AI labels versus classified labels

12.13.1 LINKAGES

This routine does not call any other subprogram. It is called by LISTSM.

12.13.2 INTERFACES

The LISTPR subprogram interfaces with other routines through the calli g arguments.

12.13.3 INPUTS

Calling sequence: CALL LISTPR(ISIT,DOTLAB,ITYPE,SUBLAB)

Parameter	Dimension	In/out	Definition
ISIT	1	In	Determines which file is printed: = 1, PPTC versus classified labels; = 2, GT versus classified labels; and, = 3, AI versus classified labels.
DOTLAB	19,11,4	In	Array containing dot labels.
ITYPE	1	In	Determines type of dot (1 or 2) input.
SUBLAB	19,11	In	Array containing subclass labels.

12.13.4 OUTPUTS

This subprogram outputs one, two, and/or three classification performance tables on the line printer.

12.13.5 STORAGE REQUIREMENTS

This subprogram requires 1444 bytes of storage.

12.13.6 DESCRIPTION

Not required.

12.13.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.13.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.14 LISTSM

The LISTSM subprogram supports LIST processing within the DISPLAY processor.

12.14.1 LINKAGES

This routine calls the CMERR, LISTLC, and LISTPR subprograms. It is called by DSPLY2.

12.14.2 INTERFACES

The LISTSM subprogram interfaces with other routines through common blocks DOTVEC, INFORM, and LISTMM and through the calling arguments.

12.14.3 INPUTS

Input to the LISTSM subprogram consists of the AI, the GT, and the PPTC dot data files output by the LIST processing system.

Calling sequence: CALL LISTSM(TOTALS,TTOL,PCTAB,LISTSW,GTUNIT,
GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,NAMECT,ALP,FLDCNT,NCAT,CATNM,
SUBCAT,NFLD2,NSUB2,SUBNAM)

Parameter	Dimension	In/out	Definition
TOTALS	1	In	Array containing total number of pixels in each subclass, including DO, DU, and thresholded pixels in the classified field.
TTOL	1	In	Array containing thresholded pixels from each classification subclass.

Parameter	Dimension	In/out	Definition
PCTAB	NOTRFD,NOSUB3	In	Array containing the classification performance table (subclass numbers for the 209 pixels in dot position by machine classification after thresholding and DO/DU editing).
LISTSW	1	In	Switch indicating selection of LIST processing.
GTUNIT	1		Number of the unit containing the GT dot data file.
GTFILE	1	In	Number of the GTUNIT file to be processed.
AIUNIT	1	In	Number of the unit containing the AI dot data file.
AIFILE	1	In	Number of the AIUNIT file to be processed.
PPUNIT	1	In	Number of the unit containing the PPTC dot data file.
PPFILE	1	In	Number of the PPUNIT file to be processed.
NAMECT	1	In	Name of small grains category which has been invoked by the category classifier.
ALP	2	In	Array containing floating-point values of α_1 and α_2 for bias correction.
FLDCNT	1	In	Number of fields to be processed (from MAPTAP file); if = 1, all dot data files are read for initialization.

Parameter	Dimension	In/out	Definition
NCAT	1	In	Number of machine categories (corresponds to NOCAT in the DISPL common block).
CATNM	1	In	Array containing category names.
SUBCAT	1	In	Map of subclasses to categories.
NFLD2	1	In	Number of training fields to be processed; set = 209 in LIST.
NSUB2	1	In	Number of subclass (not including DO, DU, and thresholded; corresponds to NOCAT).
SUBNAM	1	In	Array containing subclass names.

12.14.4 OUTPUTS

The LISTSM subprogram prints a proportion summary of labeled pixels; an n-by-n confusion matrix; and alpha value matrix; and a dot performance summary tabulating the classified and corrected percentages, variance, random sample proportion estimate, and percentage of gain for each category.

12.14.5 STORAGE REQUIREMENTS

This subprogram requires 18 744 bytes of storage.

12.14.6 DESCRIPTION

The LISTSM subprogram reads the AI, GT, or PPTC files; prepares summary information of these data versus machine classification results for type 1 and 2 dots; and computes proportion estimates with correction of the selected category for all three cases. Reports are output, as noted in section 12.14.4.

12.14.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.14.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.15 MAPHD

The MAPHD subprogram prints the header information for the classification map.

12.15.1 LINKAGES

The MAPHD subprogram does not call any other subprogram. It is called by DSPLY2.

12.15.2 INTERFACES

The MAPHD subprogram interfaces with other routines through the calling arguments.

12.15.3 INPUTS

Calling sequence: CALL MAPHD(NOCAT,CLSSYM,CATNAM,KATNO,CLSMTX, SUBNO,SUBDES,CLSVC2,NOCLS2,NOSUB2,THRSKY,THRES)

Parameter	Dimension	In/out	Definition
NOCAT	1	In	Number of categories.
CLSSYM	1	In	Array containing symbols for categories or subclasses.
CATNAM	1	In	Array containing category names.
KATNO	1	In	Array containing categories to which classes were assigned.
CLSMTX	1	In	Array containing class names.
SUBNO	1	In	Array containing subclass numbers within each class.
SUBDES	1	In	Array containing subclass names.
CLSVC2	1	In	Array containing classes to which subclasses were assigned.

Parameter	Dimension	In/out	Definition
NOCLS2	1	In	Number of classes used in processing.
NOSUB2	1	In	Number of subclasses used in processing.

THRSKY	1	In	Key for selection of thresholding procedure:
			= 1, chi-squared;
			= 2, empirical;
			= 3, user-input;
			= 4, Fisher F-distribution;
			= 0, no thresholding.

THRES	1	In	Array containing thresholds.

12.15.4 OUTPUTS

This subprogram outputs header information on the line printer or other specified unit.

12.15.5 STORAGE REQUIREMENTS

This subprogram requires 2204 bytes of storage.

12.15.6 DESCRIPTION

Not required.

12.15.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.15.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.16 PCT

The PCT subprogram builds the classification performance table for the DISPLAY processor (entry PCT) or plots the histogram of the quadratic form for empirical thresholds (entry PCTT).

12.16.1 LINKAGES

The PCT subprogram calls the FDLINT subprogram. It is called by the DSPLY2 and EMTHRS subprograms.

12.16.2 INTERFACES

The PCT subprogram interfaces with other routines through the calling arguments.

12.16.3 INPUTS

Calling sequence:

a. CALL PCT(LINUM,IR,FIELD,VERTEX,FLDSAV,PCTAB,NOFLD,SAMSTR, SAMEND,SAMINC)

Parameter	Dimension	In/out	Definition
LINUM	1	In	Line number being tested.
IR	1	In	Array containing subclass numbers to which points or pixels were assigned on the line being processed.
FIELD	5,NOFLD	In	Array containing coordinates of the rectangular area surrounding each field, including line start and end, sample start and end, and pointer into VERTEX array for field vertices; FIELD2 = training field locations; FIELD3 = test field locations.

Parameter	Dimension	In/out	Definition
VERTEX	1	In	Array containing field vertices; VERTX2 = training field vertices; VERTX3 = test field vertices.
FLDSAV	4,NOFLD	In	Array containing field information, including field name, class and subclass numbers, and number of vertices; FLDSV2 = training field information; FLDSV3 = test field information.
PCTAB	NOFLD,1	In/out	Array containing classification performance table.
NOFLD	1	In	Number of fields to process; NOFLD2 = training fields; NOFLD3 = test fields.
SAMSTR	1	In	Beginning sample number for classified field.
SAMEND	1	In	Ending sample number for classified field.
SAMINC	1	In	Sample increment used by the CLASSIFY processor.

b. ENTRY PCTT(LINUM,IR,VR,FIELD,VERTEX,FLDSAV,DSFUNC,NOFLD, SAMSTR,SAMEND,SAMINC,CON,RANGE)

Parameter	Dimension	In/out	Definition
VR	1	In	Array contain PDF's corresponding to pixels in IR.
DSFUNC	RANGE,60	In/out	Array containing histogram of the quadratic form.

Parameter	Dimension	In/out	Definition
CON	1	In	Array containing constant factors from the PDF (one for each sub-class).
RANGE	1	In	Range of the histogram.

12.16.4 OUTPUTS

The results are returned for use by the calling routine.

12.16.5 STORAGE REQUIREMENTS

This subprogram requires 2422 bytes of storage.

12.16.6 DESCRIPTION

Subprogram PCT has two entry points: PCT is called by DSPLY2 to build the classification performance table; PCTT is called by EMTHRS to plot a histogram of the quadratic form for empirical thresholding.

Both conduct a search for the number of fields which intersect the line being tested. If none is found, control returns to the calling routine. Otherwise, the fields of interest are checked, and intercepts are found utilizing subprogram FDLINT.

If the entry point is PCT, option 1 is exercised, resulting in the construction of the classification performance table, PICTAB. If the entry point is PCTT, option 2 is exercised, resulting in a check on the correctness of each pixel's classification and the plotting of a histogram of the quadratic form.

12.16.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.16.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.17 PRTPCT

The PRTPCT subprogram prints the classification performance table.

12.17.1 LINKAGES

The PRTPCT subprogram does not call any other subprogram. It is called by the DSPLAY driver routine.

12.17.2 INTERFACES

The PRTPCT subprogram interfaces with other routines through common blocks DISPL and GLOBAL and through the calling arguments.

12.17.3 INPUTS

Calling sequence: CALL PRTPCT(FLDSAV,PCTAB,NOFLD)

Parameter	Dimension	In/out	Definition
FLDSAV	4,NOFLD	In	Array containing field information; including field name, class and subclass numbers, and field vertices; FLDSV2 = training field information; FLDSV3 = test field information.
PCTAB	NOFLD,NOSUB3	In	Array containing classification performance tabulations.
NOFLD	1	In	Number of fields used in processing; NOFLD2 = training fields; NOFLD3 = test fields.

12.17.4 OUTPUTS

This subprogram outputs the classification performance table on the line printer or other specified unit.

12.17.5 STORAGE REQUIREMENTS

This subprogram requires 23 540 bytes of storage.

12.17.6 DESCRIPTION

Using data stored in the FLDSAV and PCTAB arrays and in the common blocks, the PRTPCT subprogram outputs one or more of the following classification summaries:

a. Test fields by subclass, class, and/or category.

b. Training fields by subclass, class, and/or category.

c. Test classes by class, subclass, and/or category.

d. Training classes by class, subclass, and/or category.

The parameter TSTKEY = 1 instructs the program to print data for test fields. If TSTKEY ≠ 1, data for training fields are printed.

12.17.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.17.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.18 PRTSUM

The PRTSUM subprogram prints the classification performance summary and/or ITS summary reports for the DISPLAY processor when not in the dot processing mode.

12.18.1 LINKAGES

The PRTSUM subprogram calls the SETMRG subprogram. It is called by DSPLY2.

12.18.2 INTERFACES

The PRTSUM subprogram interfaces with other routines through common blocks DISPL and GLOBAL and through the calling arguments.

12.18.3 INPUTS

Calling sequence: CALL PRTSUM(TOTALS,TTOL,FLDESC)

Parameter	Dimension	In/out	Definition
TOTALS	66	In	Array containing total number of pixels classified into each subclass, including thresholded and DO/DU pixels.
TTOL	66	In	Array containing total pixels thresholded, by subclass.
FLDESC	1	In	Field name.

12.18.4 OUTPUTS

This subprogram outputs the classification summary and/or the ITS summary report on the line printer or other specified unit.

12.18.5 STORAGE REQUIREMENTS

This subprogram requires 9156 bytes of storage.

12.18.6 DESCRIPTION

Subprogram PRTSUM begins processing by checking each input field for an existing class, subclass, or category name. If no match occurs, an ITS summary report is not printed, a message is generated, and CRPKEY is set = 0. Otherwise, if the field is classified by existing category, class, or subclass, the CRPTYP parameter is set = 1, 2, or 3, respectively.

PRTSUM continues processing by printing a classification summary for each field, including:

a. Total number of sampled points.

b. Subclass, class, or category name.

c. Number of DO/DU pixels.

d. Total points before and after thresholding and the percentage of pixels thresholded.

e. Points within each subclass, class, or category.

f. Classified field and threshold used.

g. Points thresholded by the CLASSIFY and/or DISPLAY processor.

The value of CRPKEY is then tested and, if ≠ 1, control returns to DSPLY2; otherwise, the ITS summary report is printed, including:

a. ITS name.

b. Analyst's name.

c. Procedure configuration.

d. Ground-truth acreage proportions for each classification, including "other."

e. Total pixels and their classifications, including "other," before and after thresholding.

f. Total DO/DU pixels.

12.18.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.18.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.19 REDIF3

The REDIF3 subprogram reads and analyzes supervisor control cards for the SETUP3 subprogram.

12.19.1 LINKAGES

The REDIF3 subprogram calls the CHIN, CMERR, FIND12, FLTNUM, LAREAD, NUMBER, and NXTCHR subprograms. It is called by SETUP3.

12.19.2 INTERFACES

The REDIF3 subprogram interfaces with other routines through common blocks DISPL and GLOBAL and through the calling arguments.

12.19.3 INPUTS

Calling sequence: CALL REDIF3(TSTSAV,TSTFLD,TSTVER,VDIM,GTUNIT, GTFILE,AIUNIT,AIFILE,PPUNIT,PPFILE,NAMECT,ALP,DESSAV,DESFLD, DESVER,NOFLD4)

Parameter	Dimension	In/out	Definition
TSTSAV	4200	Out	Array containing test or DO/DU field information: TSTSAV(1,N) = field name; TSTSAV(2,N) = class number; TSTSAV(3,N) = subclass; TSTSAV(4,N) = number of vertices for field N (N = 1,2,3,···,NOFLD3).
TSTFLD	5200	Out	Array containing coordinates of the rectangular area surrounding a test or DO/DU field.
TSTVER	VDIM	Out	Array containing the sample and line coordinates of the vertices for all input test or DO/DU fields.

Parameter	Dimension	In/out	Definition
VDIM	1	In	Number of locations allocated for storage of all test or DO/DU field vertices in the variably dimensioned ARRAY in SETUP3.
GTUNIT	1	Out	Number of unit containing ground-truth dot data.
GTFILE	1	Out	Number of the GTUNIT file to be processed.
AIUNIT	1	Out	Number of unit containing the AI dot data file.
AIFILE	1	Out	Number of the AIUNIT file to be processed.
PPUNIT	1	Out	Number of the unit containing the PPTC dot data.
PPFILE	1	Out	Number of the PPUNIT file to be processed.
NAMECT	1	Out	Name of small grains category which has been invoked by the category classifier.
ALP	2	Out	Array containing floating-point values of α_1 and α_2 for bias correction.
DESSAV	4,50	In/out	Array containing DO/DU field information, including field name, class and subclass numbers, and number of vertices.

Parameter	Dimension	In/out	Definition
DESFLD	5,50	In/out	Array containing coordinates of the rectangular area surrounding each DO/DU field, including line start and end and pointer into DESVER array for field vertices.
DESVER	1100	In/out	Array containing DO/DU field vertex coordinates.
NOFLD4	1	In/out	Number of DO/DU fields being processed.

The control cards relevant to this routine are given in section 12 (table 12-1) of volume II of this user guide.

12.19.4 OUTPUTS

This subprogram outputs listings of all input control card images and of user-input thresholds for all subclasses and generates error messages if a control card or field definition card image is in error or a threshold is outside the allowable range.

12.19.5 STORAGE REQUIREMENTS

This subprogram requires 8002 bytes of storage.

12.19.6 DESCRIPTION

The REDIF3 subprogram sets up the reread buffer, scans control and field definition card images, and generates a message if an error occurs. It utilizes several utility functions and/or subprograms to accomplish this task; e.g., NXTCHR for character scanning, FIND12 for locating special symbols, FLTNUM for interpreting and storing real numbers, NUMBER for locating integers, LAREAD for reading field definition card images, CHIN for calculating chi-squared threshold values, and CMERR for generating error messages.

REDIF3 sets the threshold key for either empirical, chi-squared, user-input, or Fisher F-distribution thresholding and eliminates the unused thresholding methods. It reads in user-input thresholds or calculates chi-squared distribution thresholds, if either of these options is exercised.

In reading field definition card images, it validates class and subclass names with those on the MAPTAP file and generates an error message if a discrepancy occurs. If dot processing is accepted as an option and test and training fields are input, REDIF3 rejects the input and generates an error message.

12.19.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.19.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.20 RNORM

The RNORM function calculates the chi-squared distribution for one degree of freedom.

12.20.1 LINKAGES

The RNORM function does not call any other subprogram. It is called by CHI.

12.20.2 INTERFACES

The RNORM function interfaces with other routines through the calling arguments.

12.20.3 INPUTS

Calling sequence: RNORM(X)

Parameter	Dimension	In/out	Definition
X	1	In	Input threshold value.

12.20.4 OUTPUTS

The results are returned for use by the calling routine.

12.20.5 STORAGE REQUIREMENTS

This subprogram requires 554 bytes of storage.

12.20.6 DESCRIPTION

Not required.

12.20.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.20.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.21 SETUP3

The SETUP3 subprogram locates files on the MAPTAP and the DOTFIL, analyzes control card images, and prints supervisory information for the DISPLAY processor.

12.21.1 LINKAGES

The SETUP3 subprogram calls the CMERR, FIND12, FSBSFL, NUMBER, NXTCHR, REDIF3, and WRTFLD subprograms. It is called by the DSPLAY driver routine.

12.21.2 INTERFACES

The SETUP3 subprogram interfaces with other routines through common blocks DISPL and GLOBAL and through the calling arguments.

12.21.3 INPUTS

Input to the SETUP3 subprogram consists of the MAPTAP and DOTUNT files output by the CLASSIFY and DOTDATA processors, respectively.

Calling sequence: CALL SETUP3(ARRAY,TOP,GTUNIT,GTFILE,AIUNIT, AIFILE,PPUNIT,PPFILE,NAMECT,ALP,DESSAV,DESFLD,DESVER,NOFLD4,STOP)

Parameter	Dimension	In/out	Definition
ARRAY	10 600	In	A block of working storage passed to each processor for the variable dimensioning of other arrays.
TOP	1	In	Maximum usable storage in ARRAY; TOP = 10 600.
GTUNIT	1	In/out	Number of the unit containing the GT dot data file.
GTFILE	1	In/out	Number of the GTUNIT file to be processed.

Parameter	Dimension	In/out	Definition
AIUNIT	1	In/out	Number of the unit containing the AI dot data file.
AIFILE	1	In/out	Number of the AIUNIT file to be processed.
PPUNIT	1	In/out	Number of the unit containing the PPTC dot data file.
PPFILE	1	In/out	Number of the PPUNIT file to be processed.
NAMECT	1	In/out	Name of small grains category which has been invoked by the category classifier.
ALP	2	In/out	Array containing floating-point values of α_1 and α_2 for bias corrnction.
DESSAV	4,50	In/out	Array containing DO/DU field information, including field name, class and subclass numbers, and number of vertices.
DESFLD	5,50	In/out	Array containing coordinates of the rectangular area surrounding each DO/DU field, including line start and end and pointer into DESVER array for field vertices.
DESVER	1100	In/out	Array containing DO/DU field vertex coordinates.
NOFLD4	1	In/out	Number of DO/DU fields being processed.
STOP	1	Out	If ≠ 0, control returns to the calling routine.

The control cards relevant to this routine are given in section 12 (table 12-1) of volume II of this user guide.

12.21.4 OUTPUTS

This subprogram outputs supervisory information; a table of correspondence between category names from the MAPTAP and the DOTUNT files; a summary of DOTUNT information, including dot number, sample and line numbers, type, and category; and a list of saved training and test fields (if dot processing is not being performed).

12.21.5 STORAGE REQUIREMENTS

This subprogram requires 12 350 bytes of storage.

12.21.6 DESCRIPTION

The SETUP3 subprogram begins processing by reading the first control card image and positioning the MAPTAP on the appropriate unit and file numbers. It reads the first two records of the MAPTAP file, sets base addresses for training field information, and allocates storage space for 200 test fields. REDIF3 is called to read control and field definition card images. SETUP3 then prints user options and the MAPTAP unit and file numbers. Subprogram WRTFLD is called to print field information. (See volume IV, section 12, for formats of printed options.)

SETUP3 finds the rectangular coordinates for training fields and instructs the DISPLAY processor to perform other functions, as requested on input control card images.

12.21.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.21.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.22 TINORM

The TINORM subprogram is a function that calculates an approximation to inverse normal distribution for the chi-squared and Fisher F-distribution threshold values.

12.22.1 LINKAGES

The TINORM function does not call any other subprogram. It is called by the CHIN and FISHIN functions.

12.22.2 INTERFACES

The TINORM function interfaces with other routines through the calling arguments.

12.22.3 INPUTS

Calling sequence: TINORM(ALPHA,IFLAG)

Parameter	Dimension	In/out	Definition
ALPHA	1	In	Input confidence level.
IFLAG	1	Out	Flag indicating that the threshold value has been calculated; IFLAG = 1 indicates that the threshold value is outside the allowable range.

12.22.4 OUTPUTS

The results are returned for use by the calling routine.

12.22.5 STORAGE REQUIREMENTS

This subprogram requires 2260 bytes of storage.

12.22.6 DESCRIPTION

Not required.

12.22.7 FLOW CHART

The available subprogram flow charts for this processor are provided in section 12.23.

12.22.8 LISTING

The subprogram listing is provided in volume IV, section 12.

12.23 SUBPROGRAM FLOW CHARTS

No flow charts are provided for the DISPLAY processor.